RELIABLE PRELIMINARY SPACE MISSION DESIGN
OPTIMISATION UNDER UNCERTAINTIES IN THE FRAME OF EVIDENCE THEORY

NICOLAS CROISARD

Submitted in Fulfilment of the Requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science & Engineering
University of Glasgow

Submitted in February 2013

Nicolas Croisard: *Reliable Preliminary Space Mission Design - Optimisation Under Uncertainties In The Frame of Evidence Theory*
Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy
School of Engineering, College of Science & Engineering
University of Glasgow

SUPERVISORS:
Dr. Gianmarco Radice
Dr. Massimiliano Vasile

EXAMINERS:
Dr. Geoffrey T Parks
Engineering Design Centre
University of Cambridge
Cambridge, United Kingdom

Dr. Jongrae Kim
College of Science & Engineering
University of Glasgow
Glasgow, United Kingdom

Glasgow, Scotland, United Kingdom
February 19, 2013

ABSTRACT

In the early phase of the design of a space mission it is generally desirable to investigate as many feasible alternative solutions as possible. Traditionally a system margin approach is used in order to estimate the correct value of subsystem budgets. While this is a consolidated and robust approach, it does not give a measure of the reliability of any of the investigated solutions. In addition the mass budget is typically overdimensioned, where a more accurate design could lead to improvements in payload mass. This study will address two principal issues typically associated with the design of a space mission: (i) the effective and efficient generation of preliminary solutions by properly treating their inherent multi-disciplinary elements and (ii) the minimisation of the impact of uncertainties on the overall design, which in turn will lead to an increase in the reliability of the produced results.

The representation and treatment of the uncertainties are key aspects of reliable design. An insufficient consideration of uncertainty or an unadapted mathematical representation leads to misunderstanding of the real issues of a design, to delay in the future development of the project or even potentially to its failure. The most common way to deal with uncertainty is the probabilistic approach. However, this theory is not suitable to represent epistemic uncertainties, arising from lack of knowledge. Alternative theories have been recently developed, amongst which we find Evidence Theory which is implemented in this work. Developed by Shafer from Dempster's original work, it is regarded by many as a suitable paradigm to accurately represent uncertainties. Evidence

Theory is presented and discussed from an engineering point of view and special attention given to the implementation of this approach.

Once mathematically represented, the uncertainties can be taken into account in the design optimisation problem. However, the computational complexity of Evidence Theory can be overwhelming and therefore more efficient ways to solve the reliable design problem are required. Existing methods are considered and improvements developed by the author, to increase the efficiency of the algorithm by making the most of the available data, are proposed and tested. Additionally, a new sample-based approximation technique to tackle large scale problems, is introduced in this thesis. Assuming that the uncertainties are modelled by means of intervals, the cluster approximation method, and especially implemented as a Binary Space Partition, appears to be very well-suited to the task.

The performance of the various considered methods to solve the reliable design optimisation problem in the frame of Evidence Theory is tested and analysed. The dependency on the problem characteristics, such as dimensionality, complexity, or multitude of local solutions are carefully scrutinised. The conclusions of these tests enables the author to propose guidelines on how to tackle the problem depending on its specificity.

Finally, two examples of preliminary space mission design are used to illustrate how the proposed methodology can be applied. Using realistic and current mission designs, the results show the benefits that could be achieved during the preliminary analyses and feasibility studies of space exploration.

DEDICATION

I would like to express my most sincere gratitude to Dr. Radice, without whom I would not be writing these lines today. Not only was he the one who secured the funding for my research, and who recruited me and trusted me to do the job. But he put me back on track when I was in much doubt I will ever finish to write my dissertation. And all that with a simple phone call, at the right time, using the right words of encouragement and support. For this Gianmarco, I am forever grateful.

My family deserves a big thank too. I realise that you, Mum and Dad, had a hard time because I flew away from home, but an even harder one when I was struggling with my writing. I know you felt powerless to help me, but being here and supporting me was very important to me. You were in my thoughts when I was thinking to give up. Surely you will always remember the New Year's eve of 2011! Thanks you also to my brother Sebastien, my sister-in-law Virginie, and their wonderful daughter, my most beloved niece Manon. A big hug and many kisses to the five of you.

During my years in Glasgow, I have met amazing men and women. My friends from the basketball team, the Gators, you made my Scottish experience ever so rich. It was a pleasure to play with you guys, Glen, Nicky, Cubby, Mark, Bob, Andy, Wei Hao, Big John and wee John, etc. I surely enjoyed our nights out in Sauchiehall street too. My friends from the university, Stuart, David, Giangi, Christie, Nita, Anna, Edmondo, you formed an important part of my daily life, in and out of the lab. My kitesurfing friends from Troon, in particular John and David my instructors, but also Scott, Andrew, John, Steve, etc... I

had some awesome sessions with you around Scotland, with the highlight being without a doubt our trip to Tiree.

Some of the people I met during my PhD have a special place in my life. Joan-Pau, and his lovely partner Ana, are surely amongst them. Yes, they are well-known for their heated arguments. But to me, they are full of life, passionate and just so great to be around. Matteo, aka the Italian bb, is another person I wish to thank here. He was a loyal friend of mine during all these years. How many times did he help me during my PhD? How many times did he open his door to me? How many times did he write something silly on Skype and made me smile?

I would like to especially thank Camilla, for being my unconditional friend, whether I deserved it and sometimes did not. You are a very special person indeed Cami, a little angel always on my shoulder. Supportive, full of life, always ready for anything, a smile on your face. Forever I cherish the day I met you.

To all of you, a big thank you.

Nicolas Croisard

Glasgow, February 19, 2013

ACKNOWLEDGEMENTS

This PhD was supported by the EPSRC (Engineering and Physical Sciences Research Council) and Astrium Ltd, in the frame of the Cooperative Awards in Science and Engineering (CASE), number CASE/CNA/05/61. I was fortunate enough to have Stephen Kemble of Astrium following my work. During a placement in my second year, I even had the chance to work along him at Astrium, which resulted in my research doing a major leap forward.

I would also like to acknowledge the important contribution in this work of Dr. Vasile, my supervisor at Glasgow University. His influence on this PhD thesis was significant, and I thank him for his support and many suggestions.

Finally, a thank you to Dr. Geoffrey Parks and Dr. Jongrae Kim, the external and internal examiners of my PhD viva, for your valuable comments and our enriching conversations during.

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Glasgow, Scotland, United Kingdom, February 19, 2013

Nicolas Croisard

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF SYMBOLS

The acronyms and symbols used in this thesis are presented below.

ACRONYMS

AAB	Axis-Aligned Box.
AU	Astronomical Unit.
BDR	Best Design Region.
BPA	Basic Probability Assignment.
BSP	Binary Space Partition.
CBF	Cumulative Belief Function.
CCBF	Complementary Cumulative Belief Function.
CCPF	Complementary Cumulative Plausibility Function.
CPF	Cumulative Plausibility Function.
CPU	Central Processing Unit.
ET	Evidence Theory.
JAXA	Japan Aerospace Exploration Agency (Japanese space agency).
LHSU	Latin Hypercube Sampling Uniform.
oAAB	Outer Axis-Aligned Box.
OUU	Optimisation Under Uncertainty.

PPU	Power Processing Unit.
RBDO	Reliability-Based Design Optimisation.
RBRDO	Reliability-Based Robust Design Optimisation.
RDO	Robust Design Optimisation.
s.t.	Such that.
SAA	Sun Aspect Angle.
SEP	Solar Electric Propulsion.
ToF	Time of Flight.

MATHEMATICAL OPERATORS

$()^f$	Superscript indicating the dependency on the function f.
$()_{\mathbf{d}}$	Subscript indicating the dependency on the design vector $\mathbf{d}$.
$()_{opt}$	Subscript indicating the optimality (see in the light of context).
$\widetilde{()}$	Approximation of. For instance, $\widetilde{CBF}$ is an approximation of the CBF.
$*$	Product between two scalars.
$+=$	Add to. $x+=y$ means to add y to x (used in pseudo algorithms).
$[a, b]$	Interval of reals greater than or equal to a and less than or equal to b.
$\cap$	Intersection of sets.
$\exists$	There exists.

$\forall$	For all.
$\in$	Element of.
$\leftarrow$	Assign value. $x \leftarrow y$ means assign value of y to x (used in pseudo algorithms).
$\notin$	Not an element of.
$\rightarrow$	Transform into.
$\subset$	Subset of.
$\times$	Cartesian product between 2 sets.
$\varnothing$	Empty set.
$\wedge$	And.
$O()$	"Big O notation" describing the limiting behaviour of a function.
2^{Θ}	The power set of Θ.
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}^n$	Set of n-tuples of real numbers.
$\mathcal{B}$	A box (see §3.3.3).
$\mathcal{B}_{invalid}$	A box identified as invalid.
$\mathcal{B}^{\mathbf{d}}_{invalid}$	A set of boxes identified as invalid for a specific design $\mathbf{d}$.
$\mathcal{D}$	Set of design variables.
$\mathcal{FE}$	The set of focal elements.
$\mathcal{U}$	Set of input variables.

$\mathcal{X}_{sample}$	Set of sample points $(\mathbf{d}, \mathbf{u})$ in $\mathcal{D} \times \mathcal{U}$, i.e. the design and uncertain domains combined.
$\mathcal{X}_{sample}^{y^*}$	A subset of $\mathcal{X}_{sample}$, such that all points are verifying the proposition $f(\mathbf{d}, \mathbf{u}) < y^*$.
$\mathcal{Y}$	Codomain of the system function, i.e. the set of the system function output.
$\mathcal{Y}^*$	The set of reals in $\mathcal{Y}$ that are less than or equal to the threshold y^*.
Θ	The frame of discernment.
$f^{-1}(\mathcal{Y})$	Set of input variables of the function f that return an output in the set $\mathcal{Y}$.
δy^*	Difference or step between 2 successive values of the threshold y^*.
$\mathbf{bl}$	Vector of Belief levels.
$\mathbf{bl}(i)$	i^{th} element of the vector of Belief levels.
$\mathbf{d}$	Vector of design variables.
$\mathbf{u}$	Vector of uncertain variables.
$\mu_f(\mathbf{d})$	The mean value of the system function f over the domain of uncertain parameters $\mathcal{U}$ and for a given design vector $\mathbf{d}$.
ν	Threshold of a pool function.
$\overline{A}$	The complement of A.
$\rho_{\mathbf{d}}$	Distance of the best designs found and the optimal designs.

$\min_{\mathbf{d}\in\mathcal{D}}$	Minimum such that $\mathbf{d} \in \mathcal{D}$.
A^*_{error}	Normalised error area.
A_{error}	Error area, defined as the area between the optimal Pareto front and the results found.
$Bel(A)$	Belief of proposition A.
f	Mathematical function representing the performance of a system.
f^{-1}	The inverse of the function f. If $f(x) = y$, then $f^{-1}(y) = x$.
f_{cheby}	Chebyquad function.
f^U_{cheby}	The modified Chebyquad function, used as system function of optimisation under uncertainty.
f_{pool}	Pool function of the system function f.
FE	A focal element.
g	Constraint function.
ln	Natural logarithm function.
$m(E)$	Basic Probability Assignment of the proposition E. Also referred to as mass of the proposition E.
$m_{(1\otimes2)}$	The Basic Probability Assignment resulting from the combination of 2 distinct sources of evidence, source 1 and source 2.
n_U	Number of uncertain parameters.
n_{bel}	Number of Belief levels.
n_D	Number of design variables.

n_{FE}	Number of focal elements.
n_{pop}	Population size for agent-based algorithm.
$P(A)$	Probability of proposition A.
$Pl(A)$	Plausibility of proposition A.
T_i^*	Chebyshev polynomial of order i.
y^*	A threshold.
$[\lambda_1, \lambda_2, \lambda_3]$	Shaping parameters (see [Novak and Vasile, 2009] for more details).
$[p, f, g, h, k]$	Pseudo-equinoctial elements (see [Novak and Vasile, 2009] for more details).
δ_p	Percentage of the maximal power that is wasted.
ΔV	Delta-V, change of velocity.
$\dot{m}_{sc}$	Mass flow rate, i.e. the time derivative of the spacecraft mass.
η_{engine}	Efficiency of the engine to convert electric power into thrust.
η_p	Power conversion efficiency.
κ_A	Area margin for the solar arrays.
$\kappa_{harness}$	Mass margin for the harness subsystem.
κ_{rad}	Mass margin for the radiator.
κ_{SA}	Mass margin for the solar arrays.
$\mathbf{c}$	Control vector (see §5.2.1).

μ	Gravitational constant.
$\overline{I_{SP}}$	Mean specific impulse.
$\rho_{harness}$	Specific ratio mass/power of the harness subsystem.
ρ_{SA}	Specific ratio mass/area of the solar arrays.
σ_{tank}	Specific ratio of the tank subsystem.
θ_l	Tuning parameter for Kriging surrogate model.
A_{SA}	Area of the solar arrays.
g_0	Gravitational acceleration ($g_0 = 9.80665$ m.s^{-2}).
G_s	Solar constant at 1AU ($G_s = 1367$ W.m^{-2}).
$I_{SP}{}_{max\ T}$	Specific impulse at maximum thrust.
I_{SP}	Specific impulse.
L	Longitudinal anomaly.
$m^0_{harness}$	Inevitable mass of the harness subsystem (constant).
m^0_{sc}	Launch mass of the spacecraft.
$m^0_{thrusters}$	Inevitable mass of the thrusters subsystem (constant).
$m_0{}_{array}$	Structural mass of the solar arrays (constant).
$m_{nominal}{}_{thrusters}$	Nominal mass of one thruster.
$m_{SEP}{}_{wet}$	Total SEP related mass, including propellant.
m_{array}	Mass of the solar arrays.
$m_{harness}$	Mass of the harness.

m_{PL}	Mass of the payload.
m_{PPU}	Mass of the power processing units.
m_{rad}	Mass of radiator.
m_{sc}	Mass of the spacecraft.
m_{struct}	Mass of the structure of the spacecraft.
m_{tank}	Mass of propellant tank.
$m_{thrusters}$	Mass of the thrusters.
m_{xenon}	Mass of propellant (xenon).
n_{PPU}	Number of power processing units.
$n_{thruster}$	Number of thrusters installed aboard the spacecraft.
P_{1AU}	Power to be generated by the solar arrays at 1 Astronomical Unit.
p_l	Tuning parameter for Kriging surrogate model.
P_{dis}	Dissipated power.
P_{max}	Maximum power.
Q	Heat to be dissipated at perihelion.
T	thrust.
t_0	Departure time, initial epoch of the trajectory.
T_{max}	Maximum thrust.
ToF_{viol}	Violation of the time of flight.
V	Voltage.

1

INTRODUCTION

The space industry is traditionally torn between the contrasting dichotomy of reliability and the need for performance. On one hand, sending probes to visit our celestial neighbourhood or reaching the outer edges of the solar system requires the development of high tech solutions, always trying to push the boundaries of what was achieved before. Due to the enormous cost of development, implementation and launch, the design is sought to be as efficient as possible. On the other hand, little can be done once the engines of the rocket launching the spacecraft in orbit are ignited. Additionally, the space era is still, despite its recent 60th anniversary, relatively young, and the system designs will frequently face environments which are poorly, if at all, known and understood. After all, this is the reason space exploration happens in the first place, to extend the knowledge of our world.

In this first chapter, the motivations and objectives of this study are introduced. The background of reliable design is then succinctly presented and additional information related to the scope of this thesis given. More details will be given as adequately as possible in the core chapters of the dissertation. Finally, the main contributions of the present work are summarised, and the document outline given.

1.1 RESEARCH MOTIVATIONS AND OBJECTIVES

The aim of this research is to investigate the problem of efficient and reliable space mission design. The early stages in the development of a space mission are of particular interest, as many aspects of the configuration and overall design of the spacecraft are extremely uncertain and strongly sensitive to the multitude of different choices available to the analysts and engineers.

Evidence Theory is regarded by many as a suitable paradigm to represent uncertainties. Therefore, the first objective of this study is to apply Evidence Theory in the context of reliable design optimisation. In particular, the focus here is on multi-disciplinary optimisation problems, in order to take into account for instance the trajectory as well as the propulsion or power subsystems for a space mission design.

The practical application of Evidence Theory will be discussed at length in the following. One of the main issues faced is the computational complexity associated with it. The second and principle objective is thus to study the efficiency of existing methods, and to propose alternative ones to solve robust design optimisation problems in the frame of Evidence Theory.

The third and final objective is to highlight to the space industry the benefits of using robust design methodology in general, in the frame of Evidence Theory in particular, and thus during especially the early stages of the space mission analysis and design process.

1.2 BACKGROUND

During the early phase of the design of a space mission, it is generally desirable to investigate as many feasible alternative solutions to the overall mission

architectures, as possible. At this particular stage, an insufficient consideration for uncertainty could lead to a wrong decision on the feasibility of the mission. Traditionally, a system margin approach is used in order to take into account the inherent uncertainties related to the computation of the system budgets. A system margin, also referred to as design margin or management margin, is the difference between the requirement and the demonstrated capability [Grady, 2006]. Thus, the design is consciously oversized to be robust to uncertainties. The reliability of the mission is then independently computed in parallel. An iterative, though integrated, process between the solution design and the reliability assessment should finally converge to an acceptable solution. As the development progresses, the maturity of the design increases, thus the uncertainty tends to diminish and system margins are accordingly reduced. The system margin approach is a simple and effective way to deal with uncertainty. However, defining a margin is a difficult task, and experience or tradition are frequently the sole justification of chosen value. The design obtained through this approach presents the risk of being either over conservative, thus less efficient than it could have been, or not robust enough, thus leading to costly iterations during the development.

An alternative to this approach consists in modelling the uncertainties and introducing them explicitly in the design process. The overall system design is then optimised, minimising the impact of uncertainties on the optimal values of the design criteria. The minimisation of the impact of uncertainties in the design process is generally known as robust design and the associated optimisation process robust optimisation.

The concept of robust design originates in the 1950s when Genichi Taguchi suggested engineers actively design quality into their product [Taguchi, 1978, Taguchi and Phadke, 1984, Taguchi and Yokoyama, 1993]. To do so, the design

should be such that the product's performance is insensitive to the variation in variables beyond the designer's control [Trosset, 1997]. Taguchi's *parameter design method* is a 3 stage approach: (i) the *system design* determines the feasibility region of the design, in preparation for the next step, (ii) the *parameter design* whose purpose is to optimise the objective function that quantifies the notion of quality, and finally (iii) the *tolerance design* consisting in reducing and controlling the sensitivity to critical variables [Roy, 2010].

Over the last few decades, robust design has been gaining wide attention and its applications have been extended from improving the quality of individual components to the design of complex engineering systems. Sensitivity analysis combined with safety margins are still widely used in the industry but this approach can be considered as the *worst case* philosophy. On the other hand the methods for robust design have evolved towards nonlinear optimisation formulation with multiple objectives subject to feasibility robustness [Chen et al., 1996, Phadke, 1995, Du et al., 2000] or compromise programming [Chen et al., 1998, Das, 2000, Vasile, 2002].

A key element of reliable design is how the uncertainties are mathematically represented in the design process. Uncertainties are usually classified as either aleatory or epistemic. Aleatory uncertainties are due to the random nature of input data while epistemic uncertainties are generally linked to incomplete modelling of the physical system, the boundary conditions, unexpected failure modes, etc. The traditional approach to represent uncertainty is through Probability Theory, and this has been widely used and implemented [Du et al., 2000, Du and Chen, 2002]. However, critics have highlighted some limits in applicability, because it fails to correctly and accurately represent epistemic uncertainties. [Agarwal et al., 2003, Bae et al., 2002, Hoffman and Hammonds, 1994]. A few alternatives to this approach have therefore emerged over the

last decades such as Interval Analysis [Hayes, 2003, Kreinovich et al., 2006], Possibility Theory [Zadeh, 1999], Fuzzy Set Theory [Dubois and Prade, 1989], or finally Evidence Theory [Dempster, 1966, Dempster, 1967, Shafer, 1976] considered in this thesis.

Although introducing epistemic and aleatory uncertainties in the design process would greatly improve the quality of the design (and would give a measure of the reliability of the result), it significantly increases the computational cost of any multidisciplinary optimisation. This is even more the case, if the evaluation of the cost function (and/or the constraints) associated to each discipline of a multidisciplinary problem is already computationally expensive. Implementing naively Evidence Theory is straightforward, but the untraceable computational cost associated with it has limited its usage [Barnett, 1981, Jousselme et al., 2002].

1.3 CONTRIBUTIONS

Evidence Theory is at the heart of this research. As with any young theory, it is not widely known and often not completely comprehended. A detailed and theoretical study of Evidence Theory is out of the scope of the present work. The theory is instead presented and discussed from an engineering point of view. A practical approach is taken, theoretical concepts kept as simple as possible, and special attention given to the implementation of this approach within a well defined engineering context.

Once integrated in a multidisciplinary optimisation problem, Evidence Theory's computational complexity can be overwhelming. Therefore, an efficient way to solve the design optimisation under uncertainty is required. Some have been suggested by researchers in different scientific and engineering fields and

they are discussed in this work. Improvements are introduced to the implementation of the bi-objective formulation to increase the efficiency of the algorithm by making the most of the data available. Additionally, a new sample-based approximation technique is proposed to tackle large scale problems. Assuming that the uncertainties are modelled by means of intervals, the cluster approximation method, and especially implemented as a Binary Space Partition, appears to be very well-suited to the task.

A comprehensive test campaign on an analytical test case is presented. The performance of the various considered methods to solve the reliable design optimisation problem in the frame of Evidence Theory is analysed in detail. The dependency on the problem characteristics, such as dimensionality, complexity, or multitude of local solutions are scrutinised in detail. The conclusions of these tests enables the author to propose guidelines on how to tackle the problem depending on its specificity.

Finally, two examples of preliminary space mission design are used to illustrate how the proposed methodology can be applied. Using realistic and current mission designs, it is hoped that the benefits would be more evident for preliminary analyses and feasibility studies of space exploration.

1.4 DISSERTATION OUTLINE

The core of this dissertation is composed of four chapters. The first two present Evidence Theory, the reliable design optimisation problem and how it can be tackled. The last two focus on the practical aspects of the problem at hand. The outline of the thesis is as follows.

The second chapter focuses on the mathematical modelling of uncertainties. After presenting the type of uncertainties usually encountered in the specific

case of space mission design, Evidence Theory is introduced. A parallel with Probability Theory is then drawn in order to better grasp its specificity. Engineering applications of Evidence Theory are discussed, highlighting the issues faced and suggesting ways to resolve them or to limit their impact.

The third chapter is dedicated to the reliable design optimisation problem, alternatively named Optimisation Under Uncertainty (OUU). First, the general definitions of robust and reliable design optimisation are given, and then the particularities due to using Evidence Theory are discussed. Direct (exact) and indirect (approximating) methods to solve OUU problems are then presented. The direct methods use a multi-objective optimisation algorithm, and differ following the formulation of the problem. Attention is also given to the algorithmic aspect of these methods. Finally, an approximation method through a clustering algorithm, which aims at mitigating the computation burden associated with Evidence Theory is proposed.

The forth chapter presents an exhaustive analysis of the performances of the proposed methods. The analytical test case used for this purpose, the Chebyquad problem, is first formulated. Then is presented the methodology applied to assess the performances, drawbacks and benefits of each approache to solve the OUU. After discussing the results obtained, the chapter concludes with guidelines on how to select the appropriate method based on the properties of the problem at hand and the user's aim.

Two space-related problems are then used as test case in the fifth chapter. The first problem is derived from the BepiColombo mission to Mercury. The second one simulates a feasibility study of a space mission.

Finally, the sixth chapter concludes the dissertation by summarising the findings of this thesis, and suggesting areas of further research.

1.5 PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publications.

Articles and Book Chapters

1 **Robust Preliminary Space Mission Design under Uncertainty** (Massimiliano Vasile, Nicolas Croisard), *Chapter in Computational Intelligence in Expensive Optimization Problems (Yoel Tenne, Chi-Keong Goh, Lim Meng Hiot, Yew Soon Ong, eds.)*, Springer Berlin Heidelberg, volume 2, pp. 543-570, 2010.

2 **Preliminary space mission design under uncertainty** (Nicolas Croisard, Massimiliano Vasile, Stephen Kemble, Gianmarco Radice), *Acta Astronautica*, volume 66, pp. 654-664, 2010.

Conference Papers

3 **System Engineering Design Optimisation Under Uncertainty for Preliminary Space Mission** (Nicolas Croisard, Massimiliano Vasile), *Chapter in 2009 IEEE Congress on Evolutionary Computation, Proceedings*, IEEE, volume 1-5, pp. 324-331, 2009. (IEEE Congress on Evolutionary Computation Trondheim, NORWAY, MAY 18-21, 2009)

4 **Efficient Robust Optimisation For Space Mission Design in the Frame of Evidence Theory** (Nicolas Croisard, Stephen Kemble, Mas-

similiano Vasile, Gianmarco Radice), *4th International Conference on Astrodynamics Tools and Techniques*, 2010.

5 **Semi-analytical solution for the optimal low-thrust deflection of Near-Earth Objects: Efficient robust opimisation for space mission design in the frame of evidence theory** (Nicolas Croisard, Massimiliano Vasile, Gianmarco Radice), *60th International Astronautical Congress*, 2009.

6 **Preliminary space mission design under uncertainty** (Nicolas Croisard, Massimiliano Vasile, Stephen Kemble, Gianmarco Radice), *59th International Astronautical Congress IAC 2008*, 2008.

7 **Reliable Trajectory Design Through Evidence Theory and Multiobjective Optimization** (Massimiliano Vasile, Nicolas Croisard), *Advances in Global Optimization*, 2007.

8 **Uncertainty modelling in reliable preliminary space mission design** (Nicolas Croisard, Matteo Ceriotti, Massimiliano Vasile), *Proceedings of International Joint Conference on Artificial Intelligence*, 2007.

2

MODELLING UNCERTAINTIES THROUGH EVIDENCE THEORY

This chapter focuses on the mathematical modelling of uncertainties. Over the last few decades, this problem has been considered by many researchers in different fields ranging from data fusion to decision making, from risk assessment to design optimisation. The importance of an adapted framework was stressed to avoid misrepresentation and inexact interpretation of the conclusions following the analysis performed.

After presenting the type of uncertainties usually encountered in the specific case of space mission design, Evidence Theory is introduced. This theory is considered by many as well suited to address this problem, but was nevertheless the subject of various criticisms and countless debates amongst theorists and researchers.

A parallel with Probability Theory is then drawn in order to better grasp the specificity of Evidence Theory. Finally, the last section of this chapter deals with practical considerations, and examples of engineering applications of Evidence Theory are given.

2.1 UNCERTAINTIES IN SPACE MISSION DESIGN

Uncertainties are usually classified in two distinct categories, aleatory and epistemic uncertainty. According to Helton and Oberkampf [Helton, 1997,Helton et al., 2007], the definition of each type is:

aleatory uncertainty arises from what is considered to be an inherent randomness in the behaviour of the system under study.
also known as: Stochastic uncertainty, Type A uncertainty, Irreducible uncertainty, Variability, Objective uncertainty.

epistemic uncertainty arises from a lack of knowledge about a quantity that is assumed to have a fixed value in the context of a particular analysis.
also known as: Subjective uncertainty, Type B uncertainty, Reducible uncertainty, State of Knowledge uncertainty, Ignorance.

Some researchers consider a third category, **Error**, also called numerical uncertainty, which "is defined as a recognisable deficiency in any phase or activity of modelling and simulation that is not due to lack of knowledge" [Agarwal et al., 2003]. Such uncertainties are well-known, and a good estimation of the error is generally available. This point distinguishes errors from epistemic uncertainties. Aleatory uncertainties are due to the random nature of input data while epistemic uncertainties are generally linked to incomplete modelling of the physical system, the boundary conditions, unexpected failure modes, etc.

In the case of preliminary space mission design, analysts face both types of uncertainty. For example, the initial velocity of the spacecraft, the gravity model or the solar radiation, all give rise to aleatory uncertainties. On the other hand, most of the parameters that define the characteristics of the spacecraft

subsystems are not known a priori and their values cannot be computed exactly as they hinge on other unknown parameters. Their value has therefore to be first estimated on the basis of previous experience and historical data or through educated guesses by a group of experts. The uncertainty associated to those parameters is therefore epistemic.

Traditionally, Probability Theory has been used for the modelling of uncertain parameters. However, a few alternatives to this approach have emerged over the last decades, amongst which is Evidence Theory. The reader may be interested in alternative theories such as Interval Analysis [Hayes, 2003, Kreinovich et al., 2006], Possibility Theory [Zadeh, 1999], Fuzzy Set Theory [Dubois and Prade, 1989], Theory of Paradoxical Reasoning [Smarandache and Dezert, 2005] or Theory of Clouds [Neumaier, 2004]. The article by Klir and Smith [Klir and Smith, 2001] provides a detailed analysis of the relationship between these various theories, and how they are ordered by levels of generality.

2.2 INTRODUCTION TO THE EVIDENCE THEORY

Evidence Theory was developed by Shafer [Shafer, 1976] based on Dempster's original work [Dempster, 1967]. The theory is a generalisation of classical probability and possibility theories, and has been studied and applied as it can handle both aleatory and epistemic uncertainties. It is also a well-suited framework for expert-based information, where intervals are commonly used and conflicting evidence may arise.

In this section, the fundamentals of Evidence Theory are presented. First, the basic concepts of frame of discernment, power set and basic probability assignment are defined. How sources of evidence are combined using original Dempster's original rule, and also more recent alternatives, is then discussed. In

common with many of the other theories dealing with uncertainties, Evidence Theory introduces 2 measures of likelihood: the Belief and the Plausibility. The final part of this section defines and discusses in detail these two uncertainty quantifications, and how they relate to each other.

2.2.1 *Frame of discernment* Θ*, power set* 2^Θ *and Basic Probability Assignment*

The frame of discernment Θ, also known as the universal set, is "a set of mutually exclusive elementary propositions" [Bae et al., 2002]. C. Murphy [Murphy, 2000] adds that the list of propositions should be exhaustive.

If we use Schrödinger's cat to illustrate Evidence Theory rather than quantum mechanics, then the frame of discernment is composed of only two propositions *the cat is alive* and *the cat is dead*. Clearly, these two propositions are elementary, mutually exclusive (the cat cannot be alive and dead at the same time) and exhaustive (the cat is either alive or dead, there is no other option).

The power set of Θ, 2^Θ, is the set of all subsets of Θ. It contains therefore all the possible propositions, not necessarily elementary nor exclusive, that we want to quantify. In the Schrödinger's cat example, the power set 2^Θ is composed of 4 elements, i.e. *the cat is dead*, *the cat is alive*, *the cat is either dead or alive* and the empty set $\varnothing$.

The level of confidence one has on an element E of 2^Θ is quantified using the Basic Probability Assignment (BPA) also called mass (m). A BPA satisfies the following three axioms:

$$m(E) \geq 0, \forall E \in 2^\Theta \tag{2.1}$$

$$m(\varnothing) = 0 \tag{2.2}$$

$$\sum_{E \in 2^\Theta} m(E) = 1 \tag{2.3}$$

The BPA is therefore a function that maps the power set into $[0, 1]$. The elements of 2^{Θ} are solely defined by their associated BPA, and the ones with strictly positive BPA are commonly called focal elements (FE). In the remainder of this thesis, the set of focal elements is referred as $\mathcal{FE}$, and the couple $(m, \mathcal{FE})$ as a BPA-structure.

Some particular cases of BPA-structure might be of interest [Shafer, 1976]. The term *vacuous* BPA-structure is used if the frame of discernment is the only focal element, i.e. $\Theta = \mathcal{FE}$. In this case, the uncertainty is complete. A *consonant* BPA-structure is such that the focal elements are nested, i.e. $FE_1 \subset FE_2 \subset \cdots \subset FE_n$. Finally, if all focal elements are singleton sets, then the BPA-structure is referred to as *Bayesian* as the Belief function (c.f. §2.2.4) necessarily verifies the Bayes' rule of additivity* (G. Shafer proved this in Theorem 2.8 of [Shafer, 1976]).

When more than one parameter is considered uncertain (e.g. u_1 and u_2), the power set is composed of the cartesian products of all the elements of the power sets of each parameter's frame of discernment: $2^{(\Theta_1,\Theta_2)} = 2^{\Theta_1} \times 2^{\Theta_2}$. Thus the BPA of a given element of $2^{(\Theta_1,\Theta_2)}$ is the product of the BPA of the two corresponding elements:

$$\forall (E_1, E_2) \in 2^{\Theta_1} \times 2^{\Theta_2},\ m_{1,2}(E_1 \times E_2) = m_1(E_1) * m_2(E_2) \tag{2.4}$$

A focal element of $\mathcal{FE}_{1,2} \subset 2^{(\Theta_1,\Theta_2)}$ is therefore necessarily formed by a focal element from $\mathcal{FE}_1$ and one from $\mathcal{FE}_2$. Also, the number of focal elements increases exponentially with the number of uncertain parameters and their respective number of focal elements. If n_U parameters are considered uncertain

*The Bayes' rule of additivity is: If $A \cap B \neq \varnothing$, then $Bel(A \cap B) = Bel(A) + Bel(B)$

and $|\mathcal{FE}_k|$ represents the number of focal elements relative to the k$^{\text{th}}$ uncertain parameter, the total number of focal elements in $\mathcal{FE}_{1,2,\cdots,n_U}$ is given by:

$$|\mathcal{FE}_{1,2,\cdots,n_U}| = \prod_{k=1}^{n_U} |\mathcal{FE}_k| \tag{2.5}$$

This extension is based on the assumption that the different uncertain parameters are independent. Throughout this work, we will consider this assumption to hold. Indeed, this is in a large majority the case in practical engineering applications. For instance, the efficiency of an engine, and the power to mass ratio of solar arrays are clearly independent parameters. Also, the present thesis focuses on optimisation under uncertainties, and therefore the case of dependent parameters is beyond its scope. The reader can refer to the work of [Ferson et al., 2004] for more information about this issue.

2.2.2 *Combining sources of evidence*

Evidence can come from different sources, and this information could be partially or totally conflicting. The opinion of two or more experts can be available as evidence, the results of analysis or experiments, the measurements of multiple sensors, etc. There is no reason, a priori, to consider one body of evidence and ignore the others. Evidence Theory is "based on the assumption that these sources are independent" [Sentz and Ferson, 2002]. All the available information has therefore to be combined. Many different rules of information combination exist, and depending on each particular circumstance, one or another may be preferred.

Dempster's rule is without any doubt the best known and possibly the most commonly used approach to combine bodies of evidence. G. Shafer even

wrote [Shafer, 1976] that "the heart of the theory [of Evidence] is Dempster's rule of effecting this combination". If two sources of evidence are conceptualised into two BPA functions m_1 and m_2, the combined BPA $m_{(1\otimes 2)}$ is defined by:

$$m_{(1\otimes 2)}(A) = \begin{cases} \frac{\sum_{B\cap C=A} m_1(B)*m_2(C)}{1-K} & \text{if } A \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

$$K = \sum_{B\cap C=\varnothing} m_1(B) * m_2(C) \tag{2.7}$$

This rule strongly emphasises the agreement between bodies of evidence. The conflicting ones are ignored via the normalisation factor [Sentz and Ferson, 2002]. Zadeh was one of the first to point out the issue. He wrote in 1986 that "the relational point of view leads to the conjecture that [Dempster's rule] cannot be applied until it is ascertained that the bodies of evidence are not conflicting" [Zadeh, 1986]. A number of other rules have thus been developed to deal with conflicting bodies of evidence. K. Sentz and S. Ferson have collected a wide but non-exhaustive list and classified them as follows [Sentz and Ferson, 2002]:

DEMPSTER'S LIKE RULES The Discount and Combined method [Shafer, 1976, Dubois and Prade, 1992] (a weight factor is applied to each source, the greater the confidence in a source, the larger the weight factor), Yager's modified Dempster's rule [Yager, 1987] (the conflict between sources is attributed to the universal set instead of being ignored), Inagaki's unified combination rule [Inagaki, 1991] (a unification of Dempster's and Yager's rules),

AVERAGING RULES p-Averaging [Sentz and Ferson, 2002] (a generalisation of averaging for probability distributions), Convolutive x-averaging [Sentz and Ferson, 2002] (a generalisation of the average for scalar numbers),

DISTANCE-BASED RULES Chen's new fusion approach [Chen et al., 2005] (based on a measure of the distance between conflicting sources of evidence), Zhang's centre combination rule [Zhang, 1994] (defines a measure of Belief of the intersection of two sets on which evidence is available).

A disjunctive consensus rule has also been introduced by [Dubois and Prade, 1992].

The combination rules listed here are but some of the many that have been proposed in literature. As can be seen, the analyst has many alternatives to consider, and the selection of the appropriate combination rule is a difficult issue. Moreover, as highlighted in [Oberkampf and Helton, 2002], "the results of an uncertainty analysis can strongly depend on which combination method is chosen for use". The choice of the combination rule should therefore be driven principally by the context of the information to be combined. For instance, whether the information is conflicting or not is a key point.

In this thesis, we consider only the case in which the sources of evidence have already been combined. Therefore, the aim of this subsection was not to give a comprehensive overview of the combination problem, but simply to highlight the variety of available rules and the importance of the selected one.

2.2.3 *Intervals as elementary propositions*

In most engineering applications of Evidence Theory, intervals are used as elements of the frame of discernment, i.e. as elementary propositions. To quote

P. Limbourg [Limbourg, 2005], "for the problems modelled, it is adequate (but not necessary) to restrict the [elementary propositions] to intervals rather than more complicated sets". The frame of discernment can then be viewed as the counterpart of the finite sample space in Probability Theory.

Let us consider a parameter u of which the real value is unknown. and define the associated frame of discernment as the set of 5 intervals $\Theta = \{i_1, i_2, i_3, i_4, i_5\}$ as illustrated in Figure 1. An expert then defines the BPA on

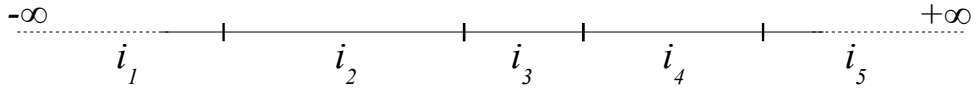


Figure 1: Example of frame of discernment based on intervals.

the elements of the power set 2^Θ by expressing his/her belief of the uncertain parameter u belonging to each of them. For example, there is a probability of 30% that $u \in i_2 \cap i_3$, 10% that $u \in i_3$, and 60% that $u \in i_4$. All other elements of 2^Θ are assigned a null BPA. Therefore, the set of focal elements is $\mathcal{FE} = \{I_1 = i_2 \cup i_3,\ I_2 = i_3,\ I_3 = i_4\}$ (c.f. Figure 2). Note that in this

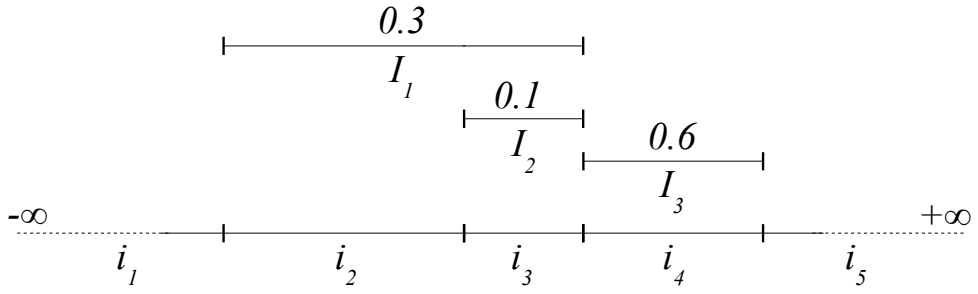


Figure 2: Example of set of focal elements based on intervals, and their associated mass.

particular example, I_1 and I_2 are not disjoint. Moreover, even though i_2 is not

a focal element, $I_1 = i_2 \cup i_3$ and $I_2 = i_3$ are two distinct focal elements, and have different mass.

When 2, 3 or more uncertain parameters are described by means of intervals, focal elements become rectangles, rectangular parallelepipeds (boxes) or orthotopes (hyper-rectangles).

In this work, intervals are unsurprisingly used throughout for the modelling of uncertainties and defining the BPA-structures.

2.2.4 *Belief and Plausibility functions*

While Probability Theory uses a single value for quantifying uncertainty, Evidence Theory uses two measures: the lower and upper bounds of the uncertainty quantification. The lower bound is called Belief (Bel) and the upper bound Plausibility (Pl) and are defined as follows:

$$Bel(A) = \sum_{\substack{FE \subset A \\ FE \in 2^{\Theta}}} m(FE) \tag{2.8}$$

$$Pl(A) = \sum_{\substack{FE \cap A \neq \varnothing \\ FE \in 2^{\Theta}}} m(FE) \tag{2.9}$$

Thus, all the propositions with a nonempty intersection with the set A contribute to the Pl value while only the propositions included in A contribute to the Bel value. For example, Figure 3 represents a BPA-structure of two uncertain parameters u_1 and u_2. Parameter u_1 is believed to belong to any of the four intervals $[a_1, b_1]$, $[b_1, c_1]$, $[c_1, d_1]$ and $[d_1, e_1]$, thus forming the set of focal elements $\mathcal{FE}_1$. Similarly, the parameter u_2 is associated with $\mathcal{FE}_2 = \{[a_2, b_2], [b_2, c_2], [c_2, d_2]\}$. Thus, the BPA-structure of the uncertain

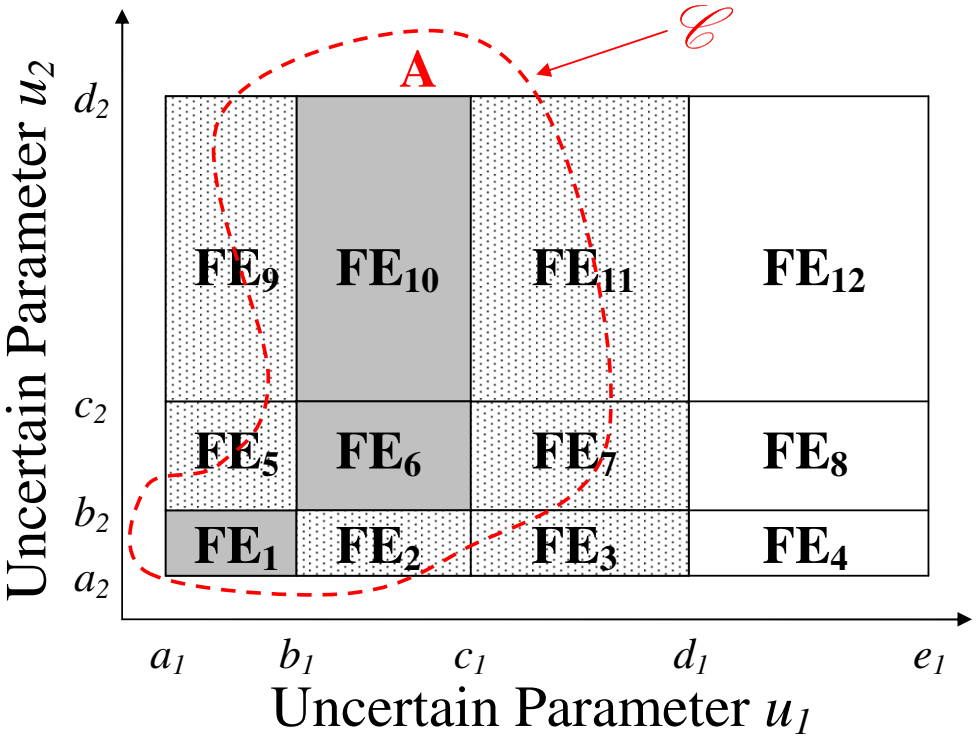


Figure 3: Belief and Plausibility of proposition A in a given BPA structure of two uncertain parameters.

domain is composed of twelve focal elements FE_1 to FE_{12}. Let us define proposition A as the area within the curve $\mathscr{C}$. Only the focal elements FE_1, FE_6 and FE_{10} (grey in the figure) are entirely included in $\mathscr{C}$. In addition, FE_2, FE_3, FE_5, FE_7, FE_9 and FE_{11} are partly inside $\mathscr{C}$ (dotted in the figure), therefore only partially implying the proposition A. The Belief and Plausibility of A are then:

$$Bel(A) = m(FE_1) + m(FE_6) + m(FE_{10})$$

$$\begin{aligned} Pl(A) = {} & m(FE_1) + m(FE_2) + m(FE_3) + m(FE_5) + m(FE_6) \\ + {} & m(FE_7) + m(FE_9) + m(FE_{10}) + m(FE_{11}) \end{aligned}$$

If the pair (u_1, u_2) takes its value within $[b_1, c_1] \times [c_2, d_2]$, it fulfils proposition A. However, if it is inside $[c_1, d_1] \times [b_2, c_2]$, it may verify A but also may not. The Belief represents then our confidence in A being always true while the Plausibility is our confidence in A being possibly true.

Three important and meaningful relations between Belief and Plausibility functions arise directly from the fact that all basic assignments must sum to 1:

$$Bel(A) + Bel(\overline{A}) \leq 1 \tag{2.10}$$

$$Pl(A) + Pl(\overline{A}) \geq 1 \tag{2.11}$$

$$Pl(A) + Bel(\overline{A}) = 1 \tag{2.12}$$

where $\overline{A}$ represents the complement of A. The two first relations show that, in contrast to Probability Theory, the Belief (resp. Plausibility) assigned to an event does not uniquely determine the Belief (resp. Plausibility) of its complement. The last relation means that Pl considers the uncertainty, while Bel does not, as shown in Figure 4.

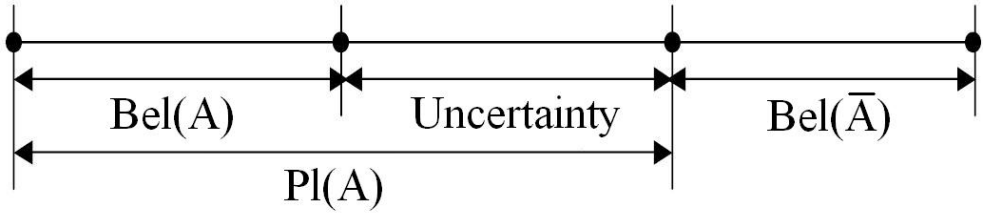


Figure 4: Interpretation of the relation between Belief, Plausibility and uncertainty (from [Agarwal et al., 2003]).

2.2.5 *Cumulative functions: CBF, CCBF, CPF, CCPF*

Analytical functions or numerical surrogates are used in engineering to represent a system. The function f, referred to as the *system function*, is defined on the

set of input $\mathcal{U}$ and return value in the codomain $\mathcal{Y}$. For the input variables $\mathbf{u}$, the quantity $y = f(\mathbf{u})$ characterises the system performance. In mathematical notation, the system function is defined as follows:

$$\begin{aligned} f &: \mathcal{U} \to \mathcal{Y} \\ \mathbf{u} &\to y = f(\mathbf{u}) \end{aligned} \tag{2.13}$$

The Tsiolkovsky rocket equation is an example of a system function, returning the change of velocity of a spacecraft for a given mass of propellant and total mass of the spacecraft. When the input variables are subject to uncertainties, the analyst is interested in propagating the uncertainties into $\mathcal{Y}$, the output domain of f. For this purpose, and similarly to Probability Theory, cumulative and complementary cumulative functions are defined in Evidence Theory to summarise the uncertainty in y.

As Evidence Theory defines two functions to quantify the uncertainty, two pairs of cumulative functions are available to the analyst: (i) the Cumulative Belief Function (CBF) and Complementary Cumulative Belief Function (CCBF) related to the Belief and (ii) the Cumulative Plausibility Function (CPF) and Complementary Cumulative Plausibility Function (CCPF) related to the Plausibility.

The CBF is defined as follows:

$$\begin{aligned} CBF &: \mathcal{Y} \to [0,1] \\ y^* &\to CBF\left(y^*\right) = Bel\left(y \leq y^*\right) = Bel\left(f^{-1}\left(\mathcal{Y}^*\right)\right) \end{aligned} \tag{2.14}$$

where $\mathcal{Y}^*$ is the set of values of $\mathcal{Y}$ that are lower than y^*, i.e. $\mathcal{Y}^* = \{y \in \mathcal{Y} \mid y \leq y^*\}$ and:

$$Bel\left(f^{-1}\left(\mathcal{Y}^*\right)\right) = \sum_{\substack{FE \in \mathcal{FE} \\ \forall \mathbf{u} \in FE,\ f(\mathbf{u}) \leq y^*}} m(FE) \tag{2.15}$$

The CBF domain of definition is $\mathcal{Y}$, the codomain of the system function f. The CBF returns values in the interval $[0, 1]$. For a threshold $y^* \in \mathcal{Y}$, the CBF of y^* corresponds to the Belief that the system function value is lower than the threshold y^*.

In the similar way, the other cumulative functions are defined below.

$$CCBF : \mathcal{Y} \to [0, 1]$$
$$y^* \to CCBF\left(y^*\right) = Bel\left(y > y^*\right) = Bel\left(f^{-1}\left(\overline{\mathcal{Y}^*}\right)\right) \tag{2.16}$$
$$CPF : \mathcal{Y} \to [0, 1]$$
$$y^* \to CPF\left(y^*\right) = Pl\left(y \leq y^*\right) = Pl\left(f^{-1}\left(\mathcal{Y}^*\right)\right) \tag{2.17}$$
$$CCPF : \mathcal{Y} \to [0, 1]$$
$$y^* \to CCPF\left(y^*\right) = Pl\left(y > y^*\right) = Pl\left(f^{-1}\left(\overline{\mathcal{Y}^*}\right)\right) \tag{2.18}$$

with the following definitions:

$$Bel\left(f^{-1}\left(\overline{\mathcal{Y}^*}\right)\right) = \sum_{\substack{FE \in \mathcal{FE} \\ \forall \mathbf{u} \in FE,\ f(\mathbf{u}) > y^*}} m(FE) \tag{2.19}$$
$$Pl\left(f^{-1}\left(\mathcal{Y}^*\right)\right) = \sum_{\substack{FE \in \mathcal{FE} \\ \exists \mathbf{u} \in FE,\ f(\mathbf{u}) \leq y^*}} m(FE) \tag{2.20}$$
$$Pl\left(f^{-1}\left(\overline{\mathcal{Y}^*}\right)\right) = \sum_{\substack{FE \in \mathcal{FE} \\ \exists \mathbf{u} \in FE,\ f(\mathbf{u}) > y^*}} m(FE) \tag{2.21}$$

Figure 5 shows an illustration of typical cumulative and complementary cumulative functions in the frame of Evidence Theory. For instance, when

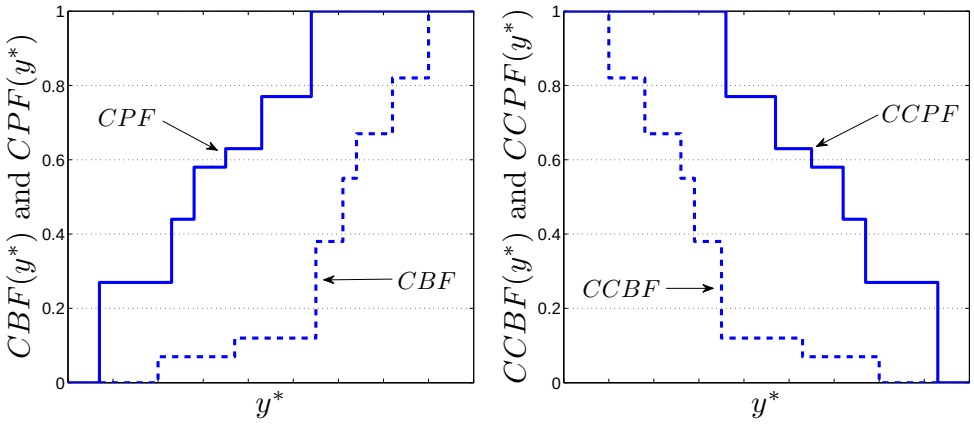


Figure 5: Examples of cumulative Belief and Plausibility functions (left) and complementary cumulative Belief and Plausibility functions (right).

$CBF\,(y^*)$ is equal to 1, there is complete certainty that the system budget remains lower the threshold y^* whatever the actual values of the uncertain parameters are. The difference between cumulative and complementary cumulative functions lies only in the sign of the inequality. The $CCBF\,(y^*)$ is the Belief that the system function value is greater than the threshold y^*, instead of lower. Similar comments apply to the CPF and CCPF.

2.3 EVIDENCE THEORY AND PROBABILITY THEORY

The classical way to treat uncertainty is through Probability Theory, but its use has been questioned over recent decades. This section provides a parallel overview of Probability Theory and Evidence Theory on 3 specific aspects: the modelling of uncertainties, the mathematical paradigm and the combining of evidence. The intention here is to highlight important differences and commonalities in these two theories, and in this way better understand

Evidence Theory through the prism of the more commonly known Probability Theory.

2.3.1 *Modelling Uncertainties*

A probability density function is well suited to mathematically model aleatory uncertainties, as long as enough data (experimental for instance) are available [Agarwal et al., 2003]. This is still the case, even when the analyst has to make assumptions on the distribution function and estimate its parameters. Moreover, [Bae et al., 2002] pointed out that aleatory uncertainty could be in fact epistemic uncertainty when "insufficient data are available to construct a probability distribution". In this situation, alternative distributions can represent the uncertainty because the mean, variance and/or shape are unknown [Hoffman and Hammonds, 1994]. The results of the analysis would therefore only reflect the arbitrary assumptions made.

Probability, however, fails to represent epistemic uncertainties because there is no reason to prefer one distribution function over another [Oberkampf and Helton, 2002]. Indeed, the probability applies only if one can identify a sample of independent, identically-distributed observations of the phenomenon of interest [Pate-Cornell, 1996]. When uncertainties are expressed by means of intervals, based on experts' opinion or limited experimental data, such as in the case of space mission design, this representation becomes questionable. As pointed out by Helton et al. [Helton et al., 2007], there is a large conceptual difference between saying that "all that is known about a quantity is that its value belongs to an interval [a,b]" and saying that "the probability distribution of that quantity is uniform on [a,b]". The latter statement, in fact, implies an

additional piece of knowledge about the distribution of that quantity on the interval [a,b].

Even though Probability Theory has been widely used to model uncertainties, including epistemic ones, it nevertheless appears to have some limitations; namely that the results and their interpretation may be questionable, as rightly illustrated in [Le Duy et al., 2010].

2.3.2 *The power set is not a σ-algebra*

The power set of the finite sample space in Probability Theory is the equivalent of the set of focal elements of Evidence Theory, in other words it is the collection of events which can be assigned probabilities. This set constitutes a σ-algebra, that is a collection of sets satisfying certain predefined properties: (i) it is not empty, (ii) it is closed under complementation and it is closed under countable union. In contrast, the set of all focal elements $\mathcal{FE} \subset 2^{\Theta}$ does not constitute a σ-algebra. This distinguishes Evidence Theory from Probability Theory on a mathematically fundamental level.

Unlike Probability Theory, unions and intersections of subsets of Θ are not necessarily focal elements: evidence on the event $\{A \textit{ or } B\}$ or $\{A \textit{ and } B\}$ does not imply/require information on either events $\{A\}$ and $\{B\}$. The set of focal elements is therefore not closed under countable union and countable intersection. This means that the *Principle of Indifference* described and criticised by Keynes [Keynes, 1921] is not needed in Evidence Theory. Moreover, the complement of a focal element is not necessarily itself a focal element. While $P(\overline{A}) = 1 - P(A)$ is true in Probability Theory, this equality does not hold in Evidence Theory.

The BPA-structure is therefore less structured than its counterpart in Probability Theory. It is designed to represent only the pieces of information available to the analyst. This characteristic is fundamental when the analyst needs to make decisions based on poor or incomplete information.

2.3.3 *Combining sources of evidence*

Probability Theory and Evidence Theory face the same problem when it comes to combining bodies of evidence, be they conflicting or not. We have already discussed the issue for Evidence Theory in §2.2.2, and listed some of the many rules available. The situation is no different for Probability Theory. A review of some aggregation methods of experts' opinions is available, for instance, in [Clemen and Winkler, 1999]. One can cite, for example, the *linear opinion pool* [Stone, 1961] as an example of mathematical combination methods or the *Delphi technique* [Rowe and Wright, 2001] for behavioural approaches.

2.4 ENGINEERING APPLICATIONS OF EVIDENCE THEORY

Evidence Theory is an efficient and interesting way to model uncertainty. However, researches on how to apply this theory to complex engineering problems have all faced significant issues. As shown previously, the choice between numerous combination rules of information is one of them. Even more problematic is the computational cost associated with this approach. This has been the most challenging difficulty in applying Evidence Theory to engineering, and many researchers have worked extensively on the subject. This section will discuss this issue, and some approximation methods proposed to deal

with it. Finally, the chapter will be concluded by presenting a few engineering applications of Evidence Theory.

2.4.1 *Computational Complexity*

Combining sources of evidence has been proven to be a #P-complete decision problem by Orponen [Orponen, 1990]. In other words, it is unlikely that a polynomial-time algorithm can solve this problem. The same point was made about computing the Belief function [Provan, 1990]. When dealing with a large number of uncertain parameters and/or numerous bodies of evidence, the computation can quickly become infeasible in practice [Burrus and Lesage, 2003].

It can also be deduced from the definitions of the cumulative functions (equations 2.14–2.18), that the computational time required to evaluate a cumulative function can become quickly prohibitive as the number of uncertain parameters and the number of intervals per parameter increase. In fact, the total number of focal elements grows exponentially with the number of uncertain parameters n_U and the number of focal elements per parameters, as seen in equation (2.5). In particular if every uncertain parameter has the same number of focal elements N, then $|\mathcal{FE}_{1,\cdots,n_U}| = N^{n_U}$. Furthermore, in order to identify the focal elements included in (or intersecting) $f^{-1}(\mathcal{Y}^*)$, the maximum of f over every focal element has to be computed and compared to y^*. In the event that the system function is convex, this maximum lies at one of the vertices of the focal element, otherwise, an optimisation problem has to be solved over every focal element. These aspects will be considered in greater detail in the next chapter, but before this, some approximation methods taken from literature on the subject are presented.

2.4.2 *Simplification of the BPA-structure*

To reduce the computational cost and make the Evidence Theory more widely usable in real world applications, approximation methods of the BPA-strucutre have been developed. The idea is to simplify it at a pre-processing stage in order to speed up the computation of the cumulative functions. Two categories of approximation are presented here.

The first and most widely studied in the literature aims at reducing the number of focal elements. Three example of approximations are detailed: the *(k-l-x)-approximation*, the *D1-approximation* and the *Inner and Outer Clustering Approximation*.

A more recent strategy consists in trying to reduce the number of uncertain parameters by evaluating their impact on the value of the cumulative function through a sampling approach. A summary of this promising but complex method is given thereafter.

k-l-x approximation

This approximation has been presented in [Tessem, 1993]. The idea here is fairly simple and can be summarised as follows:

- Keep the focal elements with higher mass
- Delete the focal elements with lower mass
- Re-normalise the mass function

By keeping the focal elements with larger mass, the loss of information is reasonably low. The re-normalisation step is only performed to ensure that the masses still add up to one. The result of the approximation depends on the 3 parameters (k, l, x). The number of focal elements kept in the approximated

mass function lies within the interval $[k, l]$. The exact number of focal elements is based on $x \in [0, 1]$ such that they represent a total mass of at least $1 - x$. The computational cost of this approximation is $O\left(n_{FE} log(n_{FE})\right)$, where n_{FE} is the number of focal elements.

D1-approximation

The D1-approximation presented by M. Bauer [Bauer, 1997] is a variant of the k-l-x method. The k elements with higher mass are still kept and the remaining ones deleted. However, the mass of the deleted focal elements is then redistributed within a selection of the k remaining ones. Following the notation used by M. Bauer, let us define, for a given integer k, the sets of focal elements M^+ and M^-:

$$M^+ = \left\{FE_1, ..., FE_k \subseteq 2^\Theta | \forall FE \notin M^+ : m\left(FE_i\right) \geq m(FE)\right\} \quad (2.22)$$

$$M^- = \left\{FE \subseteq 2^\Theta | m(FE) > 0, FE \notin M^+\right\} \quad (2.23)$$

The distribution of the mass of each focal element of M^- amongst those of M^+ is done as follow:

- The collection M_A of supersets of $A \in M^-$ is computed:

$$M_A = \left\{B \in M^+ | A \subset B\right\} \quad (2.24)$$

- The mass $m(A)$ is uniformly dispensed amongst the smallest members of M^+
- If $M_A = \varnothing$, then M'_A is constructed

$$M'_A = \left\{B \in M^+ | |B| \geq |A|, B \cap A \neq \emptyset\right\} \quad (2.25)$$

and $m(A)$ is shared amongst the smallest members of M'_A.

- The mass that cannot be distributed following the two previous steps is finally assigned to the frame of discernment Θ.

A complete description of the algorithm is available in [Bauer, 1997]. The computational time required to perform the approximation of n focal elements in k is $O\big(k,(n-k)\big)$. It has to be noted that this approximation is not well suited for elementary propositions specified by mean of intervals, as is the case in preliminary space mission design.

Inner and Outer Clustering Approximation

As a last example of an approximation based on the reduction of the number of focal elements, we present here the method proposed by T. Denoeux in [Denoeux, 2000, Denoeux, 2001]. The specificity of this approach relies on the creation of not a single but a couple of BPA $(\hat{m}^-, \hat{m}^+)$, leading to upper and lower bounds of the Pl (and in some specific cases of the Bel too). Practically, the reduction of the number of focal elements is done by regrouping successively 2 focal elements with the minimum distance ($\delta_\cap$ for $\hat{m}^-$, $\delta_\cup$ for $\hat{m}^+$) until the goal number of focal elements is achieved.

$$\delta_\cap(F_i,F_j) = m(F_i)|F_i| + m(F_j)|F_j| - \big(m(F_i)+m(F_j)\big)|F_i \cap F_j| \quad (2.26)$$

$$\delta_\cup(F_i,F_j) = \big(m(F_i)+m(F_j)\big)|F_i \cap F_j| - m(F_i)|F_i| - m(F_j)|F_j| \quad (2.27)$$

Reducing the number of uncertain parameters

An alternative to reducing the number of focal elements is to limit the number of uncertain parameters to the most influential ones. [Helton et al., 2007]

proposed a multi-step method to sequentially construct the CBF (or the other cumulative functions of Evidence Theory). The idea is essentially to first perform a sensitivity analysis to order the uncertain parameters from the most to the least contributing to the uncertainties in the system function. The CBF is constructed considering only the uncertainty of the most influential parameter, and the others are assigned degenerate evidence spaces (i.e. the sole focal element is the power set). Then, the CBF is constructed again considering 2 uncertain parameters, then 3, and so on until no meaningful changes are displayed. Thus, the analyst can decide to take into account a subset of the uncertain parameters (only the most influential ones), and assume the others completely uncertain.

2.4.3 *Examples of Engineering Application of Evidence Theory*

Many examples of application of Evidence Theory are available in literature. The intention of this section is to mention a few examples of application in engineering. [Sentz and Ferson, 2002] give as an appendix a large list of applications classified by subjects. The reader should refer to this list and the provided references for more information.

The classic scheme of application is in data fusion of multiple sensor measurements [Burrus and Lesage, 2003]. In these situations, the ease to combine bodies of evidence and the capability to deal with incompleteness or ignorance make Evidence Theory a natural choice. For instance, the authors in [Le Hégarat-Mascle et al., 1998] have at their disposal two bodies of evidence to identify forest areas. The first one is a set of optical images, which are accurate but sensitive to clouds, while the second is a set of radar images, which are less precise but not affected by cloud coverage. The aggregation of two indepen-

dent sources of evidences is illustrated in this application. Also, two types of ignorance are present in this case: imprecision of the radar images and the lack of knowledge in the optical image (when clouds cover the area of interest). Other applications related to sensor data fusion can be found in [Li et al., 2010] or [Vannoorenberghe et al., 1999].

Another area where Evidence Theory has been successfully applied is in risk assessment. For example, [Le Duy et al., 2010] used Evidence Theory to study the risk of accidents at nuclear power plants. After showing that probability theory could lead to very ambiguous or even erroneous decision making, Dempster-Shafer's theory is used as an alternative. The authors point out that "it allows decision makers to take into account the parameter uncertainty in a proper way and to have further information about the final results in the best case and worst case without having to bet on the form of a single probabilistic distribution.". Other applications linked to risk assessment and decision making are [J. Holmberg et al., 1989] or [Engemann et al., 1996].

Finally, aerospace is also a field where the Evidence Theory has been considered. For instance, the design of an Intermediate Complexity Wing is described in [Bae et al., 2002], the sizing problem of an aircraft concept in [Agarwal et al., 2003], the optimisation of robust aero-capture trajectory in [Vasile, 2004] or the preliminary design of a spacecraft in [Croisard et al., 2010].

2.5 CONCLUSIONS

Evidence Theory has been introduced and discussed at length in this chapter. It is considered as a suitable framework to modelling uncertainties. Even though it is a relatively recent theory, the theoretical foundations are well established. However, its applications are somehow limited, and mostly constrained to within

the academic community. The main reason for this lies with the computational cost associated with it. Also, engineers and decision makers in industry need to be educated about the strengths of this approach when it is best to use it, as well as how to interpret the results it provides.

Little work has been done to use Evidence Theory in reliability optimisation problems. This will be the subject of the next chapter.

3

SOLVING THE OPTIMISATION UNDER UNCERTAINTY PROBLEM

In the previous chapter, modelling uncertainties in the frame of Evidence Theory has been presented, and will be used here to formulate the Optimisation Under Uncertainty (OUU) problem. In the first section, the general definitions of robust and reliable design optimisation are given. Then the particularities due to Evidence Theory are discussed, and two practical multi-objective formulations are suggested.

The second section deals with three direct and exact methods to tackle the OUU problem. The *step method* is an attempt to solve the problem sequentially using a local optimiser. The two other methods rely on the population-based genetic algorithm. The *bi-objective method* has been introduced previously by Vasile [Vasile, 2002, Vasile, 2005]. However, no particular attention was given to the implementation. Improvements are therefore proposed here to make it more efficient.

Finally, an indirect method through a clustering algorithm is presented. This method is based on sample points and space partition, and therefore is an approximation technique. Different variants are suggested and discussed. The main objective of the indirect method is to mitigate the well-known computation burden associated with Evidence Theory, which becomes critical when the Belief is used as an objective or constraint of an optimisation problem.

3.1 THE OPTIMISATION UNDER UNCERTAINTY PROBLEM

3.1.1 *Definitions*

Solving a problem of optimisation consists in identifying the variables that minimise (or maximise) a function f traditionally referred to as the *cost function*. In engineering disciplines, it represents the system budget in the design process, and is commonly named the *system function*. The variables are then called design variables. Constant parameters are also used in the system function for the system modelling to be representative of the design problem at hand. Common system functions in space mission design are the mass of the satellite, the power consumption of the onboard subsystems and components, as well as the change of velocity necessary to fly the spacecraft to a target planet.

So the system function maps $\mathcal{D}\times\mathcal{U}$ to $\mathcal{Y}$, where $\mathcal{D} \subset \mathbb{R}^{n_D}$ is the set of possible designs $\mathbf{d} = [d_1, d_2, \ldots, d_{n_D}]$, and $\mathcal{U} \subset \mathbb{R}^{n_U}$ the set of constant parameters $\mathbf{u} = [u_1, u_2, \ldots, u_{n_U}]$. $\mathcal{Y} \subset \mathbb{R}$ is the set of possible values of f. Additionally, the design may need to verify constraints expressed via a function $g : \mathcal{D} \to \mathbb{R}$. Using this notation, a generic deterministic optimisation problem is defined as follows:

$$\begin{cases} \min\limits_{\mathbf{d}\in\mathcal{D}} & f(\mathbf{d}, \mathbf{u}) \\ s.t. & g(\mathbf{d}, \mathbf{u}) \leq 0 \end{cases} \tag{3.1}$$

If some constant parameters happen to be uncertain, as is the case during the preliminary stage of the design, then problem 3.1 is not deterministic anymore. The uncertainty of the parameters needs to be modelled, and the optimisation

formulation modified. Whether the aim is to find a robust design, a reliable design, or a reliable and robust design at the same time, the formulation will be different. First, let's recall the definition of robustness and reliability as given for example by [Yao et al., 2011]:

ROBUSTNESS The degree of tolerance of the system to be insensitive to variations in both the system itself and the environment.

RELIABILITY The likelihood that a component (or a system) will perform its intended function without failure for a specified period of time under stated operating conditions.

Note that the *Mission Reliability* can also be defined as "the likelihood that the system is operable and capable of performing its required function for a stated mission duration" [Yao et al., 2011]. The difference is that the mission reliability accepts partial failure as long as the system performs its mission, while basic reliability does not. It is therefore less stringent. Whatever the chosen definition is, the following is applicable.

Let's define $\mu_f(\mathbf{d})$ as the mean value of the system function f over the domain of uncertain parameters $\mathcal{U}$ for a given design, and $P(\cdot)$ as a measure of likelihood that a proposition is verified. We can now formulate the optimisation problem under uncertainty as follow:

ROBUST DESIGN OPTIMISATION PROBLEM (RDO) The objective is to find the optimal design point such that the likelihood of the system function being lower than a given constant $y^* \in \mathcal{Y}$, named the threshold, is maximum, while the constraints are satisfied:

$$\begin{cases} \max\limits_{\mathbf{d}\in\mathcal{D}} & P\left(f\left(\mathbf{d},\mathbf{u}\right) \leq y^*\right) \\ s.t. & P\left(g\left(\mathbf{d},\mathbf{u}\right) \leq 0\right) = 1 \end{cases} \tag{3.2}$$

RELIABILITY-BASED DESIGN OPTIMISATION PROBLEM (RBDO) The objective is to minimise μ_f, the mean value of the system function, while the likelihood of the constraints being satisfied is at least a required level $R \in [0, 1]$:

$$\begin{cases} \min\limits_{\mathbf{d}\in\mathcal{D}} & \mu_f(\mathbf{d}) \\ s.t. & P(g(\mathbf{d},\mathbf{u}) \leq 0) \geq R \end{cases} \tag{3.3}$$

RELIABILITY-BASED ROBUST DESIGN OPTIMISATION PROBLEM (RBRDO) This formulation is in fact the two previous ones combined, that is:

$$\begin{cases} \max\limits_{\mathbf{d}\in\mathcal{D}} & P(f(\mathbf{d},\mathbf{u}) \leq y^*) \\ s.t. & P(g(\mathbf{d},\mathbf{u}) \leq 0) \geq R \end{cases} \tag{3.4}$$

3.1.2 *Optimisation Under Uncertainty in the Frame of Evidence Theory*

Throughout the remainder of this chapter, we will make use of Evidence Theory to characterise the uncertain parameters. Thus, let us associate a BPA-structure to the frame of discernment $\mathcal{U}$ of the uncertain parameters $\mathbf{u}$. Using the CBF as the measure of likelihood, we can define the RBRDO as:

$$\begin{cases} \max\limits_{\mathbf{d}\in\mathcal{D}} & CBF_{\mathbf{d}}^{f}(y^*) \\ s.t. & CBF_{\mathbf{d}}^{g_1}(0) \geq R_1 \\ & CBF_{\mathbf{d}}^{g_2}(0) \leq R_2 \end{cases} \tag{3.5}$$

The subscript $()_{\mathbf{d}}$ highlights the dependency of the CBF value on the design vector $\mathbf{d}$. The superscript $()^f$ or $()^{g_i}$ identifies that the CBF is computed respectively for the system or constraint functions. The reader may wonder why problem 3.5 has two constraints rather than one as in problem 3.4. This is due to the fact that in Evidence Theory, the Belief of a proposition is not equal to 1 minus the belief of its opposite (c.f. §2.2.4). Therefore, depending on the constraint, it might be pertinent to require the Belief to be higher than (case of g_1) or lower than (case of g_2) a given level.

Note that Belief and Plausibility are interchangeable, and the choice to use either of the two for objective and/or constraints depends on the problem and the goal of the analyst. When chosen as the objective, the Belief corresponds to a strict requirement on the performance of the mission. On the other hand, if the analyst was interested in the possibility of having the mission feasible in some conditions, then Plausibility should be selected. Similar considerations are applicable to the constraints.

Although the solution to problem 3.5 gives a measure of the maximum confidence in the proposition $f < y^*$, it does not give a measure of the best achievable system budget. The simultaneous optimisation of the CBF and of f resolves this. The problem can therefore be formulated as a bi-objective optimisation problem, such that:

$$\begin{cases} \min\limits_{y^* \in \mathcal{Y}} & y^* \\ \max\limits_{\mathbf{d} \in \mathcal{D}} & CBF_{\mathbf{d}}^{f}\left(y^*\right) \\ s.t. & CBF_{\mathbf{d}}^{g_1}\left(0\right) \geq R_1 \\ & CBF_{\mathbf{d}}^{g_2}\left(0\right) \leq R_2 \end{cases} \tag{3.6}$$

A solution to problem 3.6 corresponds to a pair $(\mathbf{d}, y^*)$ such that y^* is minimal and the CBF is maximal, while the constraints are satisfied. Therefore, a pair $(\mathbf{d}, y^*)$ can be said to be Pareto optimal if there is no other pair for which the corresponding CBF is higher and y^* is lower. The image of the set of solutions that are Pareto optimal is called the Pareto front [Pareto et al., 1972].

Two sample CBF curves corresponding to two design points are represented in Figure 6. Note that depending on value of the threshold y^*, different designs can be optimal. Furthermore, the following two considerations apply to problem 3.6:

- for each value of the threshold y^*, one or more designs can maximise the belief.
- an ideal design $\mathbf{d}^*$ is such that the CBF associated to it is better than the CBF associated to any another design, for any threshold:

$$CBF^f_{\mathbf{d}^*}(y^*) > CBF^f_{\mathbf{d}}(y^*), \forall (\mathbf{d}, y^*) \in \mathcal{D} \times \mathcal{Y} \tag{3.7}$$

The latter point is particularly interesting because it defines the optimality of a set (the entire CBF curve) over another. According to this principle, the optimality of a design can be redefined as follows: a design $\mathbf{d}_1$ dominates another design $\mathbf{d}_2$ if every point in the image space corresponding to $\mathbf{d}_1$ is better, i.e has lower y^* and higher CBF, than every point in the image space corresponding to $\mathbf{d}_2$. This definition of optimality will lead us in section 3.2 to a particular formulation of the OUU, the *multi-belief approach*.

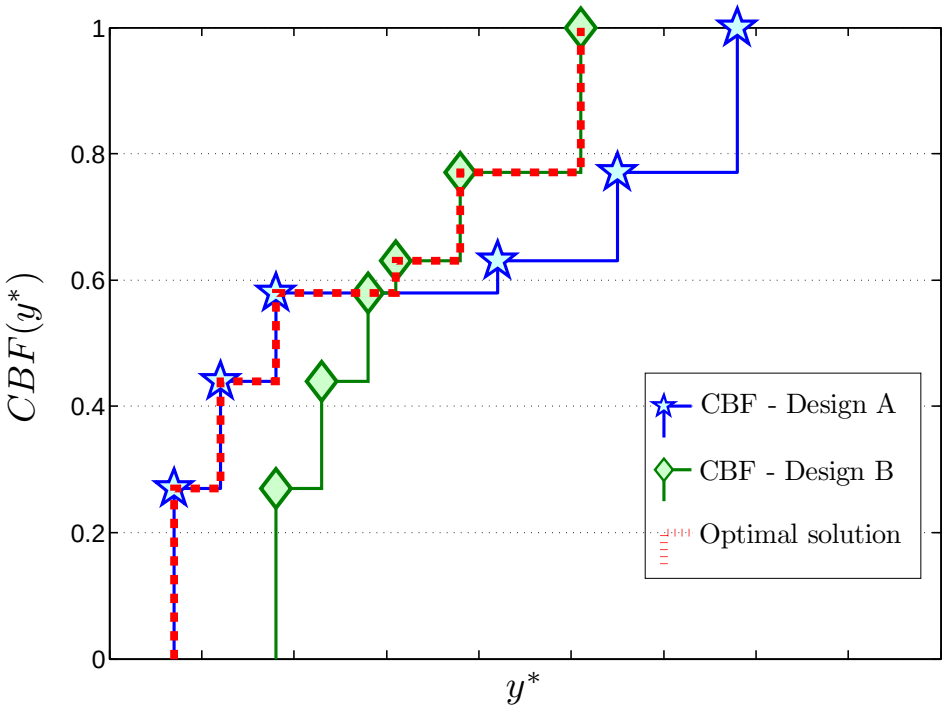


Figure 6: Typical solution of the optimisation under uncertainty problem (dash). The CBF of 2 of the dominating designs are represented ($\star$ and $\diamond$).

3.1.3 *Difficulties in Solving OUU Problems*

We have already discussed in the previous chapter, §2.4.1, the impact on the computational cost of the number of focal elements. When used in an OUU problem, as objective and/or constraints, the CPU time required to evaluate the Belief is critical. Therefore, the approximation methods presented earlier may be valuable (c.f. §2.4.2). However, finding ways to mitigate the computational cost of the OUU in the frame of Evidence Theory remains necessary. Particular attention will be given to this topic when discussing different ways to tackle OUU.

Another issue is due to the nature of the Belief and Plausibility functions. As we have seen during the presentation of the Evidence Theory in the previous

chapter, they are discontinuous. Thus, traditional gradient-based optimisers cannot handle OUU when Evidence Theory is used for modelling the uncertainties. Various solutions have been proposed to overcome this problem, such as the use of surrogate models [Agarwal et al., 2003, Agarwal et al., 2004] or multi-agent collaborative search [Scari, 2005, Vasile, 2005]. The latter is preferred in this work and will be discussed in the next section. Indeed, in addition to addressing the discontinuity issue, population-based optimisers are suitable for identifying the different designs dominating for various levels of Belief. Surrogates have been proven efficient tools on some optimisation problems with computationally expensive objective or constraints [Eldred et al., 2002]. However, great care is required to tune them to each problem, and the performance is necessarily dependent on the sample set available.

3.2 DIRECT APPROACH

The OUU problem in the frame of Evidence Theory has been formulated and discussed in the previous section. Here, direct ways to tackle the problem are presented. The focus is on the use of Evidence Theory in the objectives, but most of the comments made are applicable to constraints as well.

Three direct approaches are considered, the *step method* and two alternatives using a population-based algorithm, the *bi-objective* and *multi-belief* methods. Their respective applicability, advantages and drawbacks are discussed.

3.2.1 *Step Approach*

To be able to compute the Belief, an appropriate threshold has to be set. The step method is very straightforward as it computes the optimal belief for

discrete values of the threshold y^*. It works as follows: an initial threshold y_0^* is chosen such that a design vector $\mathbf{d}_0$, for which the Belief is equal to 1, exists. Then a smaller (or higher, depending on the problem) threshold y_1^* is selected, and a local optimiser is used to find the design $\mathbf{d}_1$ that maximises the Belief, using $\mathbf{d}_0$ as a starting point. This is repeated until a positive Belief cannot be found. The set of thresholds can be given initially by the user. Alternatively, the user fixes the step between 2 successive thresholds δy^*. A pseudo-code is presented in Algorithm 3.1.

Algorithm 3.1: Step method

Input: y^*, δy^*

Output: Matrix Out where each row corresponds to a step. The i$^{\text{th}}$ row of Out is composed of the value of the threshold, the optimum design vector and the maximum belief found at the i$^{\text{th}}$ step.

```
/* Set the initial value of the best current Belief to 1 */
```
$Bel_{max} \leftarrow 1$

while $Bel_{max} > 0$ **do**

```
   /* Update the threshold                                  */
```
   $y^* \leftarrow y^* - \delta y^*$

```
   /* Optimisation of the belief for the given threshold */
```
   $[Bel_{opt}, \mathbf{d}_{opt}] \leftarrow \max_{\mathbf{d} \in \mathcal{D}} Bel\left(f(\mathbf{d}, \mathbf{u}) < y^*\right)$

```
   /* Add a line at the end of the output matrix and save
      the results                                           */
```
   $Out(end+1, :) = [\nu, Bel_{opt}, \mathbf{d}_{opt}]$

```
   /* Update the best current Belief variable            */
```
   $Bel_{max} \leftarrow Bel_{opt}$

end

Due to the non-derivative nature of the belief function, a gradient-based optimiser is not applicable. Therefore a derivative-free algorithm (the MatLab `fminsearch` algorithm) was used. A first drawback of the step method is its dependency on the selection of the initial values of the threshold y_0^* and the threshold step δy^* (or alternatively the ordered set of thresholds). These are arbitrary choices, and at best can be informed by preliminary knowledge of

the problem at hand. If the threshold step is small, a lot of iterations will be required to solve the problem, which would be computationally expensive. On the other hand, too large a step would make the local optimisation more difficult, and the results less accurate (i.e. fewer points on the optimal Belief curve and the associated designs would have been found). A further drawback is that the procedure is sequential, which makes it impossible to have an intermediate complete Belief curve. Interrupting the algorithm prematurely (if the computational time reached a maximum for instance) would truncate the available solutions to the higher levels of Belief.

But the most important drawback of all is that this method can fail to converge to the global Pareto front or to identify multiple design points. In fact, the use of the previous optimal $\mathbf{d}_{i-1}$ to start the i^{th} iteration helps the local optimiser to converge quickly but prevents the identification of a completely different design point. Therefore, the initial design $\mathbf{d}_0$ strongly influences the convergence. In order to overcome this difficulty, a multi-start approach could be used. However, a lot of computational effort could be wasted on unnecessarily improving sub-optimal solutions or in converging multiple times to the same design.

Alternatively, a global optimiser could be selected, such as a population-based algorithm. This is presented next.

3.2.2 *Direct Solution Through a Population-based Genetic Algorithm*

Problem 3.6 is typically highly nonlinear and non-differentiable. Furthermore it can present multiple locally optimal Pareto sets, therefore for its solution we used the population-based genetic algorithm NSGA2 [Deb et al., 2002]. The approaches presented in this chapter, however, are independent of the choice

of the multi-objective optimiser. Other examples of the use of NSGA2 to solve the OUU can be found in the work of Limbourg et al. [Limbourg, 2005]. We propose here two approaches:

BI-OBJECTIVE APPROACH This approach directly tackles the bi-objective formulation problem 3.6. In this formulation, the number of objectives is limited to 2. However, the number of optimisation variables is one more than the number of design variables as the threshold is seen as both an objective and optimisation variable. Moreover, the typical solution of the OUU problem (cf. Figure 6 above) is seen in this formulation as a Pareto front.

MULTI-BELIEF APPROACH This approach consists in computing the CBF curve every time a design vector is selected. It dominates the others if there exists at least one Belief level for which its corresponding threshold is minimum. Therefore, we can see the problem as a multi-objective optimisation problem where the objectives are all the minimum thresholds corresponding to the given levels of Belief.

The two methods are discussed in more detail in the following sections.

The Bi-objective Formulation

The bi-objective formulation of the OUU problem has been applied in previous works (c.f [Vasile, 2005, Vasile, 2004]). However, no special attention was given to the actual computation of the Belief. An attempt to increase the efficiency of the algorithm is made here. The idea is to consider the focal elements in descending order of their BPA value, and to use the current best estimate of the Belief curve to speed up the computation. Algorithm 3.2 describes the proposed improvement.

Algorithm 3.2: Computing the belief in the bi-objective formulation

Inputs : y^*, $\mathbf{d}$, CBF_{opt}
Outputs: $y^*_{\mathbf{d}}$, $Bel_{\mathbf{d}}$

```
/* Initialise outputs                                              */
Bel_d ← 0;
y*_d ← −∞;
/* Identify the current optimal belief corresponding to y*         */
Bel_opt ← CBF_opt(y*) ;
/* Initialise local variables                                      */
i ← 1 ;                                              /* Counter    */
n_FE ← numel(FE) ;                  /* Number of focal elements    */
achBel ← 1 ;                        /* Achievable CBF value        */
/* Main loop                                                       */
while achBel ≥ Bel_opt and i ≤ n_FE do
    /* Compute the maximum of f on the i^th focal element          */
    y_max ← max_{u∈FE_i} f(d, u) ;
    /* Update the achievable CBF value or the outputs              */
    if y_max ≤ y* then
        Bel_d += m(FE(i));
        y*_d ← max(y_max, y*_d);
    else
        achBel −= m(FE(i));
    end
    i += 1 ;                                /* Increase counter    */
end
```

CBF_{opt} represents the current best estimate of the optimal solution in the objective space. It can be initialised to the CBF of any design, or simply set equal to the null function. When evaluating an agent a_i, corresponding to a pair $(\mathbf{d}, y^*)$ (i.e. a design point and a new threshold), CBF_{opt} is initially used to identify the current best Belief for the threshold y^*. The algorithm then starts to consider the focal elements one by one.

At every iteration, the achievable level of Belief is tracked (variable *achBel* in Algorithm 3.2). It corresponds to the hypothetical Belief of design $\mathbf{d}$ if all remaining focal elements were to be valid. Therefore, the achievable Belief is reset to 1 for each new agent. Every time a focal element is not valid (f is above y^*), the achievable Belief decreases by the corresponding BPA. As soon

as the achievable Belief is lower than $CBF_{opt}(y^*)$, the design is guaranteed not to improve the current solution, and the computation is prematurely stopped. Sorting the focal elements by descending order of their BPA is suggested to tackle first the ones that might influence most the achievable Belief.

Furthermore, once a value is assigned to the threshold y^*, the maximisation of the system function f over each focal element is stopped as soon as a value is found above the threshold. Finally, if the design is found to dominate the current solution, the minimum threshold y_d^* and the corresponding Belief level $Bel_{\mathbf{d}}$ are returned, and used to update the CBF_{opt}.

Note that to make the CBF_{opt} available throughout the computation, a global variable was used in our implementation, and was updated directly in the objective function. Therefore no modification of the genetic algorithm was necessary.

The computational cost of Algorithm 3.2 is dictated by n_{FE} the number of focal elements. However, the proposed enhancements introduced here are expected to increasingly reduce the number of system function evaluations as convergence is approached. This will be investigated in chapter 4.

The Multi-belief Approach

It is necessary for the multi-belief method to select a set of Belief levels to define the objectives for the global optimiser. Once the BPA-structure of the problem is defined, the complete set of all possible levels of Belief can be determined. While this is the preferable choice, it can lead to excessively large Belief level sets. The formulation of problem 3.8 allows the selection of only a subset and this might be a wiser choice. Also, the belief levels could be selected independently from the BPA-structure, and the algorithm would still work.

If we name $\mathbf{bl}(i)$ the i$^{\text{th}}$ chosen level of Belief, and n_{bel} the number of levels, we have to solve the following n_{bel}-objectives optimisation problem:

$$\begin{cases} \min\limits_{\mathbf{d}\in\mathcal{D}} \; y^*_{\mathbf{d}}\left(\mathbf{bl}(1)\right) \\ \min\limits_{\mathbf{d}\in\mathcal{D}} \; y^*_{\mathbf{d}}\left(\mathbf{bl}(2)\right) \\ \qquad\vdots \\ \min\limits_{\mathbf{d}\in\mathcal{D}} \; y^*_{\mathbf{d}}\left(\mathbf{bl}(n_{bel})\right) \end{cases} \tag{3.8}$$

where $y^*_{\mathbf{d}}\left(\mathbf{bl}(k)\right) = min\left(y^* \mid CBF_{\mathbf{d}}(y^*) = \mathbf{bl}(k)\right)$ correspond to the minimal threshold for which the Belief at design $\mathbf{d}$ is $\mathbf{bl}(k)$. All the n_{bel} minimal thresholds for a given design are known as soon as the entire belief curve is computed, which is done each time a design is selected.

In the case of the multi-belief approach, an agent a_i is simply $a_i = \mathbf{d}$. For each selected design vector the complete belief curve is computed. Though this is more computationally expensive than computing a single belief value, it has the benefit of having only the design vector as an optimisation variable. Therefore, each design needs to be evaluated once and only once. Additionally, in Algorithm 3.2, the known extrema of f over all focal elements evaluated during the loop are lost. Thus, while the information was available, it is not used to identify if the current design is dominating for lower belief levels (or identically lower thresholds). By computing the whole belief curve instead we preserve this information.

A more elegant implementation of this approach would consist in redefining the dominance index. If the classical Pareto dominance index

$$I_i = \left|\left\{j \mid CBF_{\mathbf{d}_j}(y^*_j) > CBF_{\mathbf{d}_i}(y^*_i) \wedge y^*_j < y^*_i,\; j = 1,\ldots,n_{pop} \wedge j \neq i\right\}\right| \tag{3.9}$$

is used to define the Pareto optimality of a design vector $\mathbf{d}_i$, where $|.|$ denotes the cardinality of a set, the optimiser cannot evaluate correctly the local Pareto optimality of a point on the $CBF - y^*$ plane since for each design there is a whole curve of points in the $CBF - y^*$ plane. If the Pareto dominance index were defined as in equation 3.10 below

$$I_i = n_{bel} - \left| \left\{ k \in [i, n_{bel}] \mid \forall j \in [1, n_{pop}],\ y_i^*(\mathbf{bl}(k)) > y_j^*(\mathbf{bl}(k)) \right\} \right| \quad (3.10)$$

then a design with a dominance index lower than n_{bel} dominates all the others for at least one of the belief levels **bl**. Therefore leading to the same result as the formulation of equation (3.8) and the standard dominance index. However, this requires the implementation of the dominance index to be modified within the optimiser, which may be complex and not always possible.

3.3 THE CLUSTER APPROXIMATION METHOD

3.3.1 *Presentation*

The direct computation of the Belief and Plausibility curves for every feasible design point can be a computationally very expensive operation, due to the complexity in the calculation of the cumulative functions even after one of the approximation techniques is applied (c.f. §2.4.2). To mitigate this, an indirect method, based on sample points of the system function, is proposed here. Sampling-based methods have been previously proposed to compute the cumulative belief function of Evidence Theory [Helton et al., 2006, Helton et al., 2007]. The cluster approximation method is partly based on this work. However,

the method proposed here is simplified to be suitable for OUU problems. Indeed, Helton et al. focused on computing the CBF for a single design.

The idea is to identify at first, within the cartesian product of the uncertain parameters domain and the design domain, the set $f^{-1}(\mathcal{Y}^*)$ where the system function verifies the proposition $f < y^*$. For a design vector $\mathbf{d}$, an approximation $\widetilde{CBF}_{\mathbf{d}}(y^*)$ of the cumulative belief function at the threshold y^* can then be cheaply computed by adding the mass of the focal elements included in any element of $f^{-1}(\mathcal{Y}^*)$ or union of elements:

$$\widetilde{CBF}_{\mathbf{d}}(y^*) = \sum_{\substack{(d,FE)\subset\cup s_i \\ s_i\subset f^{-1}(\mathcal{Y}^*)}} m(FE) \tag{3.11}$$

Alternatively, the approximation can be made by subtracting the mass of the focal elements intersecting any element of the complement set $f^{-1}\left(\overline{\mathcal{Y}^*}\right)$:

$$\widetilde{CBF}_{\mathbf{d}}(y^*) = 1 - \sum_{\substack{(d,FE)\cap s_i \neq \varnothing \\ s_i\subset f^{-1}\left(\overline{\mathcal{Y}^*}\right)}} m(FE) \tag{3.12}$$

This indirect method is referred to as the *cluster approximation method*, and is illustrated in Figure 7. In this example, there is one uncertain parameter and one design variable, respectively represented along the x and y axis. The BPA is composed of only three focal elements, FE_1, FE_2 and FE_3. The set of subdomains where the system function verifies the proposition $f \leq y^*$ is $f^{-1}(\mathcal{Y}^*) = \{s_1, s_2, s_3\}$, represented as the shaded areas.

Two different designs $\mathbf{d}_1$ and $\mathbf{d}_2$ are represented. The approximations of CBF for the two designs $\mathbf{d}_1$ and $\mathbf{d}_2$ are respectively:

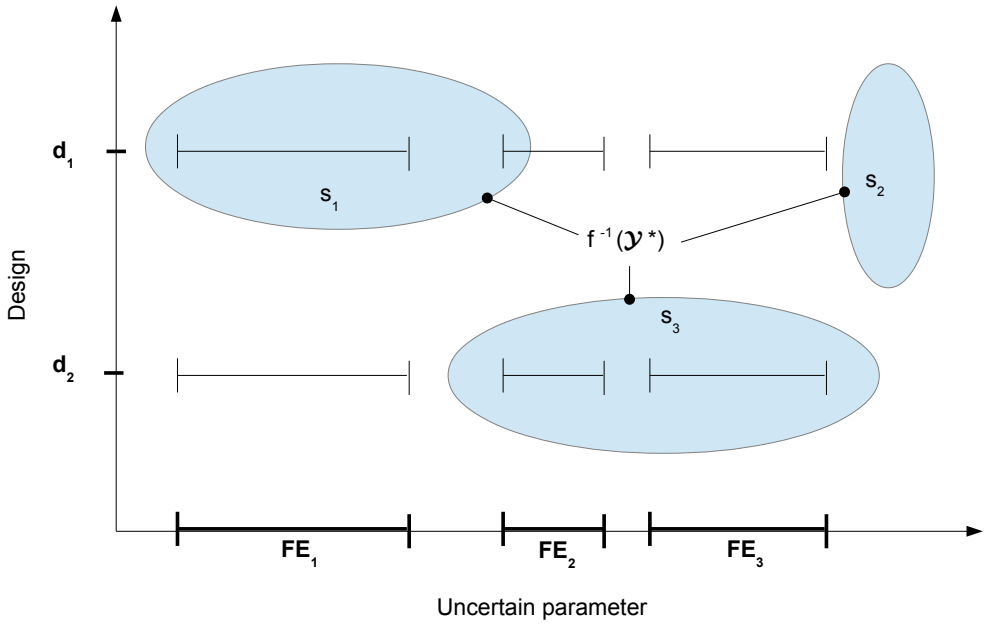


Figure 7: Illustration of the cluster approximation method with 3 focal elements FE_1, FE_2 and FE_3. The proposition $f < y^*$ is true only within the subdomains s_1, s_2 and s_3. Two examples of design points $\mathbf{d}_1$ and $\mathbf{d}_2$ are given.

$$\widetilde{CBF}_{\mathbf{d}_1}(y^*) = m(FE_1) \tag{3.13}$$

$$\widetilde{CBF}_{\mathbf{d}_2}(y^*) = m(FE_2) + m(FE_3) \tag{3.14}$$

To compute the approximation of the CBF function, the set $f^{-1}(\mathcal{Y}^*)$ is computed for increasing values of the threshold until a belief of 1 is found. As for the step method, an initial threshold y_0^* is defined, and associated with the threshold step δy^*. A set of sample points $\mathcal{X}_{sample}$ is also needed. At each step, the subset $\mathcal{X}_{sample}^{y^*}$ of sample points verifying the proposition $f(\mathbf{d}, \mathbf{u}) < y^*$ are identified, then arranged in clusters. The points of a given cluster define a subdomain s_i in $f^{-1}(\mathcal{Y}^*)$. Then, the design maximising the approximation of $\widetilde{CBF}(y^*)$ is selected. The algorithm used here is described in Algorithm 3.3.

Algorithm 3.3: Basic algorithm of the cluster approximation method

Inputs : y^*, δy^*, $\mathcal{X}_{sample}$

Output : Matrix Out where each row corresponds to a step. The i$^{\text{th}}$ row of Out is composed of the value of the threshold, the optimum design vector and the maximum approximated cumulative belief found at the i$^{\text{th}}$ step.

`/* Initialise` $\widetilde{Bel}_{max}$ `*/`

$\widetilde{Bel}_{max} \leftarrow 0$

`/* Main loop */`

while $\widetilde{Bel}_{max} < 1$ **do**

`/* Update the threshold */`

$y^* \leftarrow y^* - \delta y^*$

`/* Find the set of valid sampled points */`

$\mathcal{X}_{sample}^{y^*} \{(\mathbf{d}, \mathbf{u}) \in \mathcal{X}_{sample} \mid f(\mathbf{d}, \mathbf{u}) \leq y^*\}$

`/* Identify the valid subdomains */`

Partition in *clusters* the sample points of $\mathcal{X}_{y^*}^{valid}$

$f^{-1}(\mathcal{Y}^*) \leftarrow \{s_i\}$

`/* Find the design point giving the highest` $\widetilde{CBF}$ `*/`

$[\widetilde{CBF}_{opt}(y^*), \mathbf{d}_{opt}] \leftarrow \max_{\mathbf{d} \in \mathcal{D}} \widetilde{CBF}_{\mathbf{d}}(y^*)$

`/* Add a line at the end of the output matrix and save the results */`

$Out(end+1, :) = [y^*, \widetilde{CBF}_{opt}(y^*), \mathbf{d}_{opt}]$

`/* Update the optimum belief variable */`

$\widetilde{Bel}_{max} \leftarrow \widetilde{CBF}_{opt}(y^*)$

end

One significant advantage of this method is that it identifies all the local optimum design regions thus highligthing different classes of interesting design (as in the direct solution). The global optimum is also likely to be found using a simple local optimiser, starting for instance from the barycentre of each cluster.

The framework of the clustering approximation method has been presented. An important step in the algorithm is how the set of valid sample points is

partitioned into clusters, and then how the subdomains s_i are defined. This is the critical part of the method, as it strongly affects the results and the computational effort required to compute the approximated Belief curve. The next section introduces 3 proposed implementations.

3.3.2 *Identifying the Set of Feasible Subdomains*

Cluster, Convex hulls and Axis-Aligned Boxes

This first approach uses a clustering algorithm to classify in small subsets the valid sample points. Over the years, various clustering algorithms have been proposed and used in different fields such as chemistry [Stanforth et al., 2007] or image processing [Comaniciu and Meer, 2002]. A detailed analysis of clustering algorithms is beyond the scope of this thesis. For more information, the reader can refer to [Mirkin, 2005].

One cluster is considered representative of a single valid subdomain in $f^{-1}(\mathcal{Y}^*)$. Once the clustering phase is completed, the clusters of points are treated independently and sequentially, and used to define the boundaries of the valid subdomains. To do so, various choices are again available. A convex boundary would be attractive for the current application. Indeed, a focal element is convex itself. Therefore verifying that a focal element is included within s_i only requires a check that all its vertices are. A convex hull is a candidate that comes naturally to mind, as it is the smallest convex domain containing the valid sample points. If the convex hull has a complex shape, then checking if a point is contained in it or not may require some computational effort. This leads us to the third part of this method.

To speed up the computation, Axis-Aligned Boxes (AABs) are used. Each subdomain s_i is associated with its outer AAB (called also the Axis-Aligned

Boundary Box) and an inner AAB. If s_i is defined by the cluster of points $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}\}$ of $\mathbb{R}^L$ (here, $L = n_D + n_U$, i.e. the number of dimensions of the design vector space and uncertain parameter space combined), then its axis-aligned boundary box $oAAB(s_i)$ is defined as:

$$oAAB(s_i) = \left\{\mathbf{x} \in \mathbb{R}^L \mid \forall k, 1 \leq k \leq L, \min_{1 \leq j \leq L} x_j(k) \leq x(k) \leq \max_{1 \leq j \leq L} x_j(k)\right\} \tag{3.15}$$

The inner AAB is an axis-aligned box that is contained within the subdomain s_i. Unlike to the outer AAB, the definition of the inner AAB is not unique. It is possible, for instance, to centre the inner AAB on the barycentre of the sample points and to maximise its relative size such that it remains within s_i.

The idea behind the inner and outer AABs is that it is extremely cheap to check if a focal element is outside or inside an AAB. The focal elements that are outside the outer AABs are guaranteed not to be included in any subdomain in $f^{-1}(\mathcal{Y}^*)$, while the ones inside the inner AABs are guaranteed to be included in one subdomain in $f^{-1}(\mathcal{Y}^*)$. Once this selection process is done, only the focal elements that do not enter in any of those categories need to be checked to compute $\widetilde{CBF}(y^*)$.

In order to identify if any of the remaining focal elements fulfil the proposition $f(\mathbf{d}, \mathbf{u}) < y^*, \forall \mathbf{u} \in FE$, one only needs to check if its vertices are within the same subdomain s_i. If s_i is a convex hull, the phase 1 of the revised simplex method used to find a feasible solution to a linear programming problem can

be used [Dantzig, 1965, Bunday, 1984]. Indeed, for a point $\mathbf{v}$ of $\mathbb{R}^L$ and the convex hull $\mathcal{H}_{conv}$ of the points $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_p}\}$, we have:

$$\mathbf{v} \in \mathcal{H}_{conv} \Longleftrightarrow \exists \boldsymbol{\lambda} \in \left(\mathbb{R}^+\right)^p \mid \left(\mathbf{v} = \sum_{k=1}^{p} \lambda(k) * \mathbf{x_k}\right) \wedge \left(\sum_{k=1}^{p} \lambda(k) = 1\right) \quad (3.16)$$

The use of the simplex method as a convex hull inclusion test is detailed in [Bailey and Cowles, 1987], in which it is shown that "the average time for a single inclusion test is roughly proportional to the size of the point set times the dimensionality".

It is important to highlight that in this method, no assumptions are made on the convexity of the system function f. Only the subdomains s_i are considered as convex. For a large number of system functions, this should be acceptable. Should this not be the case, alternative ways to define the boundaries of the subdomains in $f^{-1}(\mathcal{Y}^*)$ can be envisaged. [Bates and Wynn, 2004] applied Hilbert bases to identify points on the boundary and then used surrogate such as Kriging [Sacks et al., 1989] to model the boundary. Note that if the boundary is not convex, the inclusion test of focal elements within a subdomain becomes much more complex.

Pixelisation as an Alternative

The clustering and convex hull technique presented above is suitable if the number of sample points is reasonable and if the number of dimensions is limited to just a few. Indeed, the computational time and memory requirements would quickly hamper the clustering and impact even more severely on the creation of the convex hulls. Thus, the benefits arising from the use of the approximation method would be substantially reduced.

In these cases, an alternative to identify the subdomains s_i is based on the partition into pixels of the cartesian product of the design domain $\mathcal{D}$ and the uncertain parameters domain $\mathcal{U}$. A pixel is nothing other than an axis-aligned box. No intersection is allowed between any two pixels, and the union of all the pixels equals the whole domain $\mathcal{D} \times \mathcal{U}$. This pixelisation technique replaces the use of both the clustering algorithm and convex hulls to define the boundaries of the subdomains s_i.

This is done by creating first the list of the pixels containing sample points verifying the proposition $f(\mathbf{d}, \mathbf{u}) < y^*$, then pruning this list by eliminating the pixels containing at least one sample point violating the proposition. It can be proven that this operation is polynomial with the number of dimensions and subdivisions of each dimension. A focal element is hereafter said to be valid if all the pixels intersecting it are included in any s_i.

The quality of this approximation technique is obviously related to the quality of the sampling, but also on the number and size of the pixels. The larger the pixels the lower the accuracy of the coverage and therefore the results, but the faster the algorithm. With respect to the convex hull, this approach has the main advantage that it can represent a highly non-convex subdomain s_i. Moreover, as the design domain is discretised, a finite number of different designs are accessible. Therefore, one can consider testing them all to identify the best one(s). If not, an optimiser working with binary variables can be used to solve the OUU.

Binary Space Partition

The Binary Space Partition (BSP) was first introduced by [Fuchs et al., 1980] for computer graphics. It is proposed here as an alternative method to partition the domain into regions where the proposition $f < y^*$ is verified, and regions

where it is not, based solely on sample points. Similarly to the pixelisation technique described earlier, the BSP divides the domain into hyper-rectangles, but does not require the user to specify their size nor location. Instead, the partition is done following a simple recursive division of the domain, building a full binary tree, i.e. a tree in which every node other than the leaves has two children. The leaves of the tree are the equivalent to the pixel of the previous technique, and are axis-aligned boxes like the pixels.

The idea of partitioning the domain iteratively has been applied in [Mourelatos and Zhou, 2006], where plausibility was used as constraint of a RBDO problem. However, sample points were not used, and neither were binary trees. Algorithm 3.4 presents how the binary tree is structured, in other words how the domain is partitioned and the set of valid subdomains identified. Basically, the algorithm starts with the root which corresponds to the whole domain, containing the whole set of sample points available. It is then split in two equal (in size) partitions along the first dimension. These partitions constitute the nodes of the first level. Both partitions are then split again along the second dimension to create the nodes of the second level, and so forth. A node is split only if it contains at least one sample point verifying the proposition, and another one contradicting it. Therefore, there are three possibilities for a node to become a leaf:

1. It contains only sample points fulfilling the proposition
2. It contains only sample points not verifying the proposition
3. It does not contain any sample points

It is considered here that a leaf (an axis-aligned box) is valid only in the first case. It is tagged invalid in the two others. If the depth of the tree is not limited, all the leaves are either valid or invalid. However, it might be relevant

to limit the depth of the tree, to limit the number of hyper-rectangles to one, or to limit the CPU time spent building the tree. In this case, some leaves will be undetermined, and should be considered as invalid.

3.3.3 *Inclusion Test for Focal Element*

We have discussed already in §3.3.2 how a focal element is tested once the set of valid subdomains has been defined using the convex hull. How this is done when hyper-rectangles are used (pixelisation or BSP method) remains to be defined. This is the purpose of this section. Note that the size of these boxes is assumed to be too small to include any focal element. If that was the case, a simple test on the vertices of the focal elements would suffice.

Once the domain has been partitioned in axis-aligned boxes, computing the belief is fairly straightforward. The first step is to list all the boxes which are invalid or empty, referred to as $\mathcal{B}_{invalid}$. Then, for a given design point $\mathbf{d}$, the set $\mathcal{B}^{\mathbf{d}}_{invalid}$ of boxes intersecting the design is identified. A focal element will be said valid (and therefore its mass considered as part of the belief) if its intersection with the set $\mathcal{B}^{\mathbf{d}}_{invalid}$ is empty. Therefore, the boxes of $\mathcal{B}^{\mathbf{d}}_{invalid}$ are tested one after the other until one is found intersecting the focal element. If that is the case, the check is stopped. If no intersection is found, the focal element is valid, and the associated basic probability assignment added to the Belief. The procedure is then restarted for the next focal element.

The classic technique to test the intersection of two convex polytopes, used typically in computing geometry (e.g. computer games or collision detection), is to apply the separating axis theorem [Eberly, 2000]. The theorem states that "if two convex objects are not intersecting, there exists an axis for which the projection of the objects will not overlap". When the objects are like in

our application 2 axis-aligned boxes AAB_1 and AAB_2, applying the theorem consists simply of checking if there exists a dimension for which the lower bound of ABB_1 is greater than the upper bound of ABB_2, or for which the lower bound of ABB_2 is greater than the upper bound of ABB_1. If such a condition is verified, then $AAB_1 \cap ABB_2 = \varnothing$.

3.4 CONCLUSIONS

In this chapter, Optimisation Under Uncertainty has been defined and discussed when Evidence Theory is used for modelling the uncertain parameters. Three direct exact methods have been presented, namely the step method, the bi-objective method and the multi-belief method. The step method appears to be too limited and unlikely to solve OUU problems. The bi-objective and the multi-belief methods rely on a population-based genetic algorithm, and differ in the problem formulation used. Particular attention has been given to the algorithm computing the Belief in the bi-objective method.

Additionally, an indirect method has been proposed to mitigate the computational cost of the Belief, a critical aspect when used as objective in an optimisation problem. The method consists in partitioning the domain of definition of the system function, based on sample points, into valid and invalid regions. The Binary Space Partition appears to be particularly well suited to the task. Indeed, it does not rely on any tuning, it is fast and simple, and capable of representing any shape of subdomains. Finally, testing the validity of focal elements requires little computational effort.

In the next chapter, the proposed methods will be extensively tested on an analytical test case, and their respective performances, strengths and weaknesses discussed.

Algorithm 3.4: Algorithm of Binary Space Partition (BSP) - buildTree

Inputs : p the parent node, $\mathcal{X}^p$ the set of sample points contained in the parent's domain, $\mathcal{Y}^p$ the values at the sample points

```
/* Check the status of the parent                                    */
if p.status == undetermined then
    /* Create two children                                           */
    p.left ← (new) child;   p.right ← (new)child
    /* Get the dimension along which to split the parent's
       domain                                                        */
    dim_split ←
    /* Compute the value of the middle of the domain                */
    Val_split ←
    /* Split the sample points between the children                 */
    for i = 1 to n_pts do                          /* for each point */
        /* Check if the i^th point belongs to the left child
           */
        if X^p(i, dim_split) < val_split then
            /* Assign the i^th point to the left child              */
            X^p.left ← X^p.left ∪ X^p(i, :);   Y^p.left ← Y^p.left ∪ Y^p(i)
        else   /* the i^th point is in the right child's domain
        */
            X^p.right ← X^p.right ∪ X^p(i, :);   Y^p.right ← Y^p.right ∪ Y^p(i)
        end
    end
    /* Update the status of both children                           */
    foreach child of parent do
        if There is no point in the child then
            c.status ← invalid
        else if All points in the child do not verify the proposition then
            c.status ← invalid
        else if All points in the child verify the proposition then
            c.status ← valid
        else
            c.status ← undetermined
        end
    end
    /* Split both children (recursive call)                         */
    buildTree(p.left, X^p.left, Y^p.left)
    buildTree(p.right, X^p.right, Y^p.right)
end
```

4

ANALYTICAL TEST CASE: CHEBYQUAD

Various methods have been presented in the previous chapter to solve optimisation under uncertainty problems in the frame of Evidence Theory. To investigate their strengths and weaknesses, they have been used on a set of analytical test cases. This chapter presents these findings, and aims at defining some guidelines on which method to use depending on the problem properties, the computational power available and what the analyst is really after.

In the first section, the test cases are presented. They are based on the Chebyquad function, which takes its name from the Chebyshev polynomials. The various optimisation under uncertainty problems differ in number of design variables, uncertain parameters or focal elements. Thus, the influence of these fundamental characteristics on the performance will be addressed.

The second section deals with the test campaign approach. Particular attention is given to how the solvers have been tuned. Also, the metrics used to assess the results are defined.

The results are then presented in the following two sections. Section 3 focuses on the direct methods while section 4 on the indirect ones. The Binary Space Partition, associated with two different sampling methods, is used. The pixelisation method is not tested due to its similarities with the BSP. The Convex hull has been also omitted, primarily due to its strong dependence on the cluster algorithm. Additionally, the computational cost and memory needs of the convex hull are tremendous on the most complex test cases.

Finally, in the fifth section the quality of the solutions obtained with the different methods are compared, and conclusions are drawn.

4.1 THE CHEBYQUAD OUU PROBLEM

The Chebyquad test function is used throughout this chapter as the system function of an OUU problem. The formulation of the problem follows the one given in chapter 3. The Chebyquad function is discussed in the next subsection while the following two deal successively with the BPA structure and the optimal solutions.

4.1.1 *The Chebyquad function*

Definition

The Chebyquad function, introduced by [Fletcher, 1965], is used here as an analytical test case of optimisation under uncertainties. It is of interest because its dimensionality can be chosen. This will enable us to investigate the influence of the number of design variables and uncertain parameters on the performance of the different methods presented earlier in solving the OUU problem. Moreover, the Chebyquad function presents multiple minima. This characteristic helps to create test cases for which the optimal design point changes from one confidence level to the other (c.f. §4.1.2).

The Chebyquad function is based on shifted Chebyshev polynomials $\{T_i^*, i \geq 0\}$. They are defined as follows:

$$\begin{cases} T_0^*(t) = & 1 \\ T_1^*(t) = & 2t - 1 \\ T_k^*(t) = & (4t-2) * T_{k-1}^*(t) - T_{k-2}^*(t) \forall k \geq 2 \end{cases} \tag{4.1}$$

Using these polynomials, one can define the set of functions $\{\Delta_i, 1 \leq i \leq n\}$, n being the dimension of the variable $\mathbf{x}$:

$$\begin{array}{rcl} \Delta_i : [0,1]^n & \rightarrow & \mathbb{R} \\ \mathbf{x} & \rightarrow & \Delta_i(\mathbf{x}) = \displaystyle\int_0^1 T_i^*(t)\mathrm{d}t - \frac{1}{n}\sum_{j=1}^{n} T_i^*(x_j) \end{array} \tag{4.2}$$

It can be shown that the integral of the shifted Chebyshev polynomials over the interval $[0,1]$ are:

$$\int_0^1 T_i^*(\mathbf{x})\mathrm{d}\mathbf{x} = \begin{cases} 0 & \text{if } i \text{ is odd} \\ -1/(i^2-1) & \text{if } i \text{ is even} \end{cases} \tag{4.3}$$

Finally, the Chebyquad function can be defined as follow:

$$\begin{array}{rcl} f_{cheby} : [0,1]^n & \rightarrow & \mathbb{R} \\ \mathbf{x} & \rightarrow & f_{cheby}(\mathbf{x}) = \displaystyle\sum_{i=1}^{n} \left(\Delta_i(\mathbf{x})\right)^2 \end{array} \tag{4.4}$$

Figure 8 gives a representation of the Chebyquad function for $n = 2$ while Figure 9 corresponds to $n = 3$.

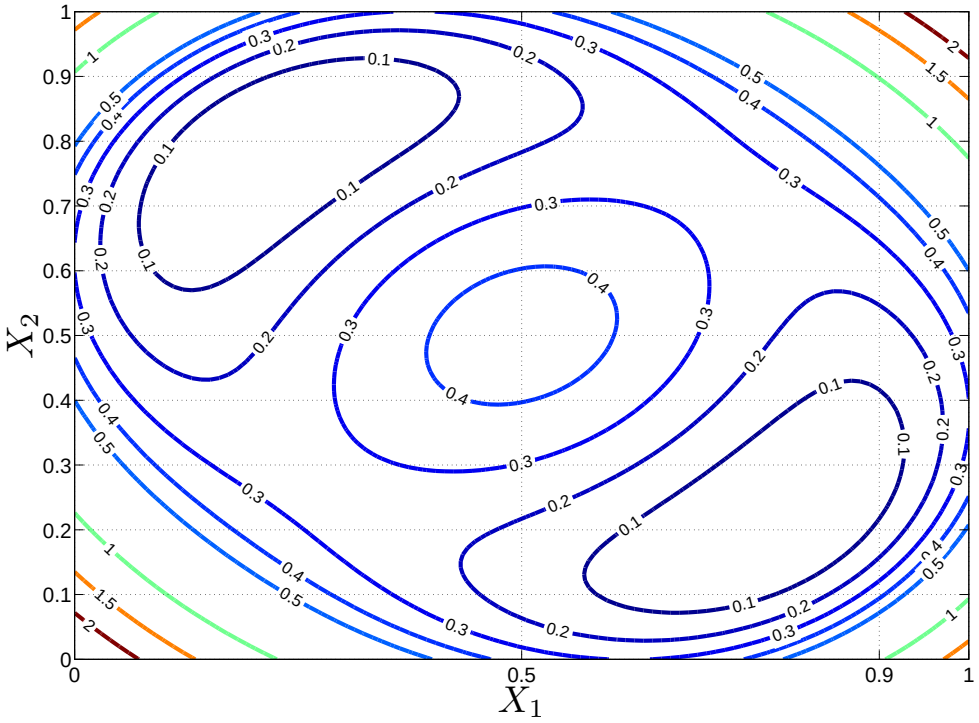


Figure 8: Contours of the Chebyquad function of 2 variables.

Evaluating the function

Evaluating the Chebyquad function (Eq. 4.4) can be done efficiently using the algorithm suggested by [Fletcher, 1965]. Note that there is a typographical error in the original code, corrected* in Algorithm 4.1.

A symmetric function

The Chebyquad function is symmetric: its value at any n-tuple of inputs is the same as its value at any permutation of that n-tuple. For the case of two variables, this translates into:

$$f_{cheby}(x_1, x_2) = f_{cheby}(x_2, x_1), \forall (x_1, x_2) \in [0, 1]^2 \tag{4.5}$$

*the line *f:=delta×delta; ieven:=false;* in [Fletcher, 1965] was replaced by *delta:=delta/n; f:=delta×delta; ieven:=false;*

Algorithm 4.1: Calculating the Chebyquad function

Inputs : x, n
Outputs: f_{cheby}

```
/* Compute Δ_1 and the shifted Chebyshev polynomials          */
```
$\Delta \leftarrow 0$
for *j=1* **to** n **do**
  $y[j] \leftarrow 2 * x[j] - 1$
  $\Delta \leftarrow \Delta + y[j]$
  $Ti[j] \leftarrow y[j]$
  $Ti\_minus[j] \leftarrow 1$
end
$\Delta \leftarrow \Delta / n$

```
/* Compute the initial value of f_cheby                       */
```
$f_{cheby} \leftarrow \Delta * \Delta$

```
/* Main loop                                                  */
```
for *i=2* **to** n **do**
  `/* Compute Δ_i and the shifted Chebyshev polynomials      */`
  $\Delta \leftarrow 0$
  for *j=1* **to** n **do**
    $Ti\_plus \leftarrow 2 * y[j] * Ti[j] - Ti\_minus[j]$
    $\Delta \leftarrow \Delta + Ti\_plus$
    $Ti\_minus[j] \leftarrow Ti[j]$
    $Ti[j] \leftarrow Ti\_plus$
  end
  $\Delta \leftarrow \Delta / n$
  `/* Subtract the value of the integral                     */`
  if *i is even* **then** $\Delta \leftarrow \Delta + 1/(i * i - 1)$
  `/* Update the value of f                                  */`
  $f_{cheby} \leftarrow f + \Delta * \Delta$
end

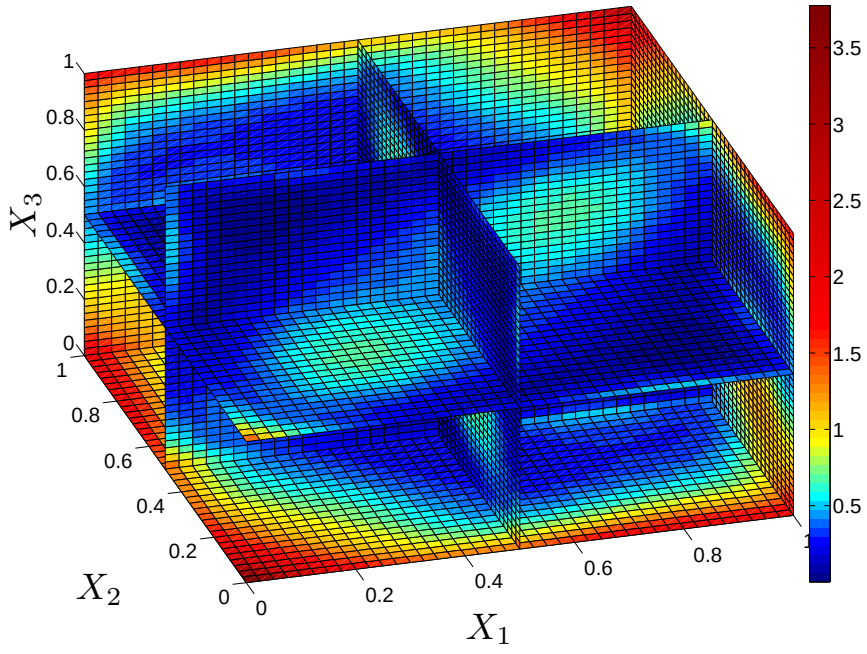


Figure 9: Slices of the Chebyquad function of 3 variables.

This property presents an issue when used as a test case for an optimum solver. Finding the minimum (or maximum) of a symmetric function is easier because there is not one but instead $n!$ points[†] for which the function is minimum. If using a population-based algorithm for instance, the density of the population is effectively much higher. If a genetic algorithm is used, the crossover will necessarily lead to offspring with ranking as high as their parents. One could say that the search space is virtually reduced, its size effectively divided by $n!$. This could create difficulties while interpreting the effect of the number of variables on the performance of the optimiser. Additionally, the symmetry penalises greatly indirect methods, which are based on sampling.

To remedy this matter, we propose to modify slightly the Chebyquad function by adding a small offset favouring one area of the search space. But the offset

[†] $n! = 1 * 2 * \ldots * n$, where n is here the number of variables of the objective function.

is only applied to the vector of design variables $\mathbf{d}$, and not to the vector of uncertain parameters $\mathbf{u}$. The following equations define the modified Chebyquad function:

$$\begin{cases} f^U_{cheby}(\mathbf{u},\mathbf{d}) = & f_{cheby}([\mathbf{u},\mathbf{d}]) + 2 * \delta f_{cheby}(\mathbf{d}) \\ \delta f_{cheby}(\mathbf{d}) = & \frac{\sum_{i=1}^{n_D} n_D^{i-1} * idx_{\mathbf{d}}(i)}{\sum_{i=1}^{n_D} n_D^{i-1} * (n_D+1-i)} - 1 \end{cases} \tag{4.6}$$

In the previous equations, $\mathbf{u}$ and $\mathbf{d}$ are respectively the vector of uncertain parameters and design variables. $[\mathbf{u},\mathbf{d}]$ denotes the concatenation of $\mathbf{u}$ and $\mathbf{d}$. Finally, $idx_{\mathbf{d}}$ is the vector of indices returned by any sorting algorithm applied to the variable $\mathbf{d}$ (`Quicksort` [Hoare, 1962] is used in our implementation). For example, if $\mathbf{d} = (0.3, 0.4, 0.05)$, then $idx_{\mathbf{d}} = (2, 3, 1)$. The offset is null when the elements of $\mathbf{d}$ are sorted in descending order. Thus the minimal value of the original Chebyquad function remains unchanged. The offset is also always null when only one design variable is used ($n_D = 1$).

In the remainder of this chapter, the modified Chebyquad function f^U_{cheby} is used as the system function of an OUU problem.

4.1.2 *The BPA structure*

For the sake of simplicity, the BPA for each uncertain parameter is identical and given in Table 1

Let us have a look at the case with one uncertain parameter ($n_U = 1$) and one design variable ($n_D = 1$), i.e the bi-dimensional Chebyquad. In Figure 10, the variation of the system function is given for 3 different designs $d_1 = 0.2$,

Lower bound	Upper bound	BPA
0	0.5	0.70
0.5	0.9	0.30

Table 1: BPA structure for each uncertain parameter in the Chebyquad test case.

$d_2 = 0.7$ and $d_3 = 0.65$. We are interested in the Belief of the proposition $f_{cheby}(u, d) < y^*$.

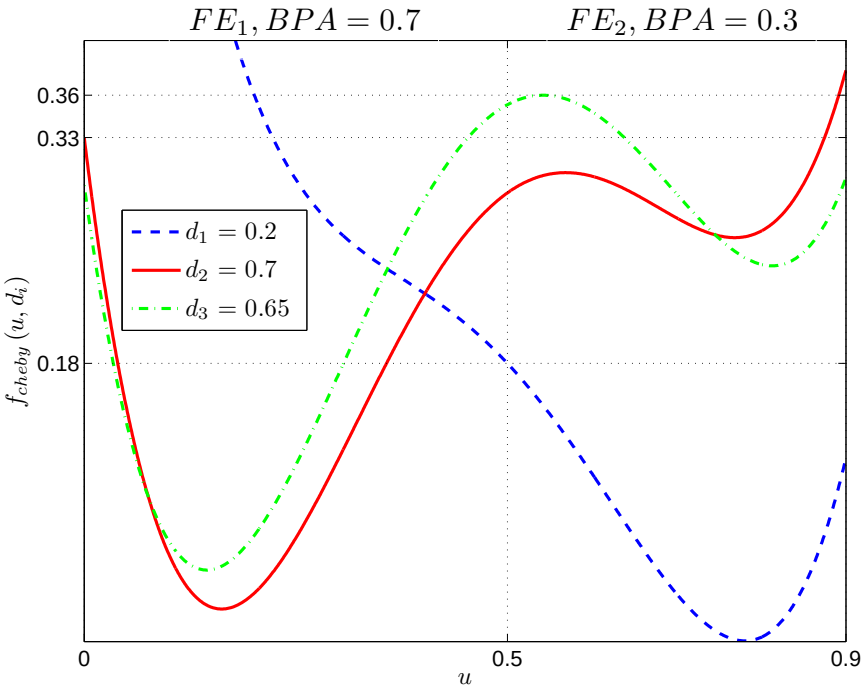


Figure 10: Chebyquad test case - Example of 3 different designs dominating for 3 different levels of Belief.

For a threshold $y^* = 0.18$, the design d_1 leads to a Belief of 0.3 (only the second focal element FE_2 is valid). If the threshold increases to $y^* = 0.33$, the same design still gives a Belief of 0.3; but a Belief of 0.7 is found at design d_2 (only the first focal element FE_1 is valid). Finally, for a threshold of $y^* = 0.36$,

only the third design d_3 sees both focal elements verified. Thus a Belief of 1 is found.

This illustrates that the optimal design will be found in distinct regions depending on the threshold level, or similarly the Belief level. The BPA structure has been designed explicitly to give the Chebyquad test case this property.

Finally, some cases will be run with larger sets of focal elements, in order to investigate the impact on the performance of the proposed methods. Other BPA structures with 3, 4 and 5 intervals per uncertain parameter are thus defined in Tables 2, 3 and 4 respectively. When these extended BPA structures are used, it is clearly mentioned. The simplest BPA with 2 intervals per uncertain parameter is the default one.

Lower bound	**Upper bound**	**BPA**
0	0.25	0.20
0.25	0.50	0.50
0.50	0.90	0.30

Table 2: BPA structure for each uncertain parameter in the Chebyquad test case - 3 intervals per uncertain parameter.

Lower bound	**Upper bound**	**BPA**
0	0.25	0.20
0.25	0.50	0.50
0.50	0.70	0.20
0.70	0.90	0.10

Table 3: BPA structure for each uncertain parameter in the Chebyquad test case - 4 intervals per uncertain parameter.

Lower bound	Upper bound	BPA
0	0.25	0.20
0.25	0.35	0.35
0.35	0.50	0.15
0.50	0.70	0.20
0.70	0.90	0.10

Table 4: BPA structure for each uncertain parameter in the Chebyquad test case - 5 intervals per uncertain parameter.

4.1.3 *The optimal solution*

The optimal designs and corresponding Pareto front of the Chebyquad test cases are not known a priori. However, aggregating the solutions of numerous attempts to solve the problem should lead to a very accurate approximation. For illustration purposes, consider a method having only a 5% chance of finding the optimal solution. Given that each trial is independent, a hundred successive runs would give a 99% probability of success, that is $1 - (1 - 0.05)^{100} = 99\%$.

Based on this remark, the results of all simulations obtained with the direct method (bi- and multi-objective) are combined. Note that the results obtained from the indirect methods are not used directly. In contrast to the direct methods, they return approximated solutions. Thus, the exact Cumulative Belief Function (CBF) of the returned designs is computed a posteriori, and then aggregated to the best results found by the direct methods. The obtained Pareto front and list of optimal designs are then used in this chapter to assess the performance of the different methods.

The list of optimal design for two test cases are given in the Tables 5 and 6. All the cases considered in this work are listed in the appendix §A. As desired (c.f. §4.1.2), the set of optimal designs is composed in all cases of at least 2 very distinct elements.

Range of Belief Levels		Design Vector
0.3	0.3	0.173842
0.7	0.7	0.682178
1	1	0.673443

Table 5: Optimal designs for the Chebyquad test case – $n_D = 1$, $n_U = 1$.

Range of Belief Levels		Design Vector		
0.0081	0.2646	0.699464	0.354749	0.125163
0.2664	0.3402	0.799063	0.487448	0.154594
0.3409	0.3483	0.45426	0.252255	0.132735
0.349	0.7518	0.757333	0.451786	0.105358
0.7522	0.7599	0.721529	0.441935	0.105807
0.7606	1	0.819988	0.462537	0.103319

Table 6: Optimal designs for the Chebyquad test case – $n_D = 3$, $n_U = 4$.

Also, the optimal Belief curve is given in Figure 11 for the cases with 1 design variable, and in Figure 12 for the cases with 3 design variables

Finally, the optimal Belief curves when the number of focal elements is changed (for $n_D = 3$ and $n_U = 2$) are presented in Figure 13. It is worth noting the influence of the number of focal elements on the curve. As the number of focal elements increases, the optimal Belief curve tends to move towards lower threshold values, but the threshold giving a Belief of one remains unchanged. This is understandable by considering the definition of Belief. Additionally, the more focal elements, the more continuous the curve appears to be. If the number of focal elements were to tend toward infinity, the curve would converge toward the cumulative density function of probability theory.

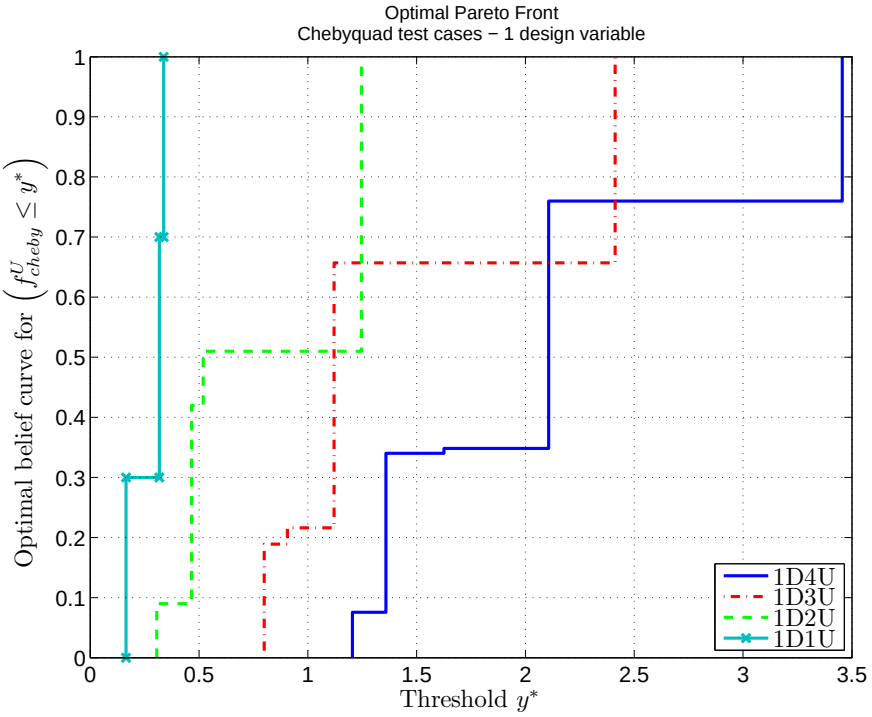


Figure 11: Optimal Belief curves for the Chebyquad test cases with 1 design variable.

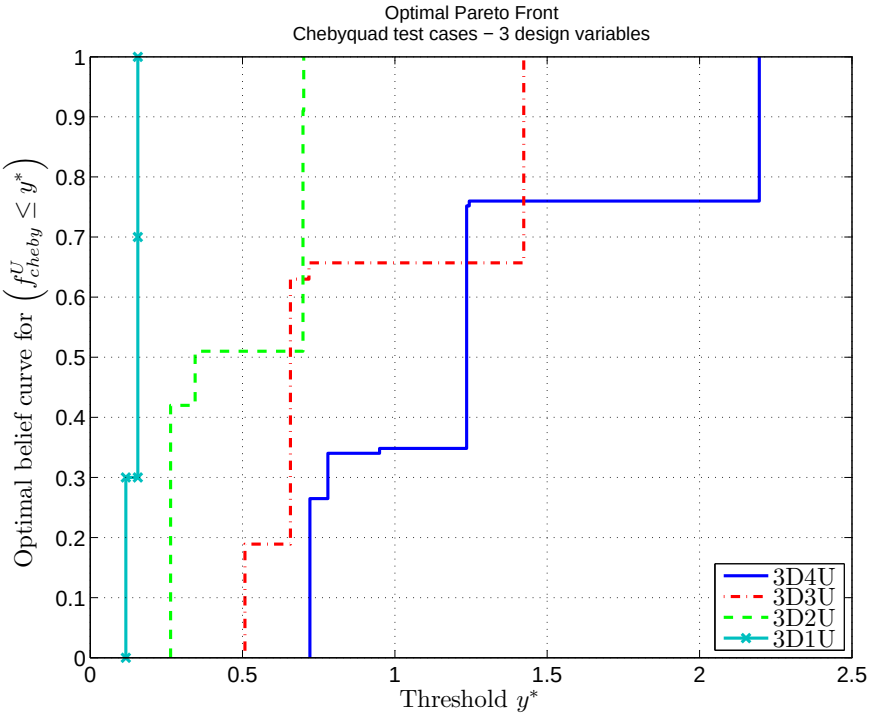


Figure 12: Optimal Belief curves for the Chebyquad test cases with 3 design variables.

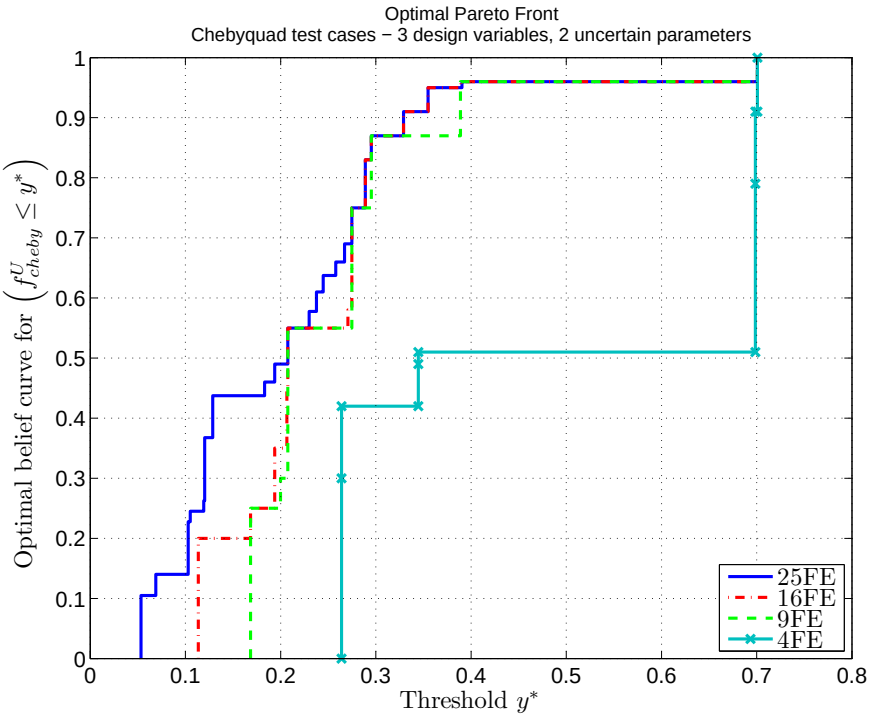


Figure 13: Optimal Belief curves for the Chebyquad test cases with 3 design variables, 2 uncertain parameters, and various numbers of focal elements.

4.2 METHODOLOGY FOR ASSESSING THE PERFORMANCE OF THE PROPOSED METHODS

This section presents the methodology used to assess the performance of the methods proposed in Chapter 3 to solve the OUU. The aim of the work presented here is to provide insights into their strengths and weaknesses. We focus particularly on the impact of the complexity of the problem on the performance of the methods. This is further explained in §4.2.1. The intrinsic performance of the global optimiser is out of the scope of this thesis. However, setting it up properly is required to compare fairly the bi-objective and multi-belief methods to the others. Similarly, other methods have tuning parameters requiring attention too. What settings have been used, and how they have been chosen, is dealt with in §4.2.2. Finally, comparison implies metrics, which are presented in §4.2.3.

4.2.1 *Varying the complexity of the problem*

By complexity, we mean the number of design variables, the number of uncertain parameters, and the number of focal elements. For Evidence Theory to be used on reliability problems, it is critical that a nearly optimal solution is found in a reasonable amount of time. As previously shown, the number of operations increases exponentially with the number of focal elements. Also, if the number of variables increases, finding a good solution becomes more difficult.

Thus, in the current assessment, the number of designs variables and uncertain parameters is varied. As explained in §4.1, the system function used here can accept any number of variables.

Additionally, for a specific number of design variables and uncertain parameters, the number of focal elements is increased. This does not change the dimensionality of the problem, but rather increases the effort required to evaluate the CBF of each design. As a consequence the number of possible optimal levels of Belief increases too.

Design variables	**Uncertain parameters**	**Focal elements per uncertain parameters**	**Focal elements**
1	1	2	2
	2	2	4
	3	2	8
	4	2	16
2	2	2	4
3	1	2	2
	2	2	4
	3	2	8
	4	2	16
3	2	3	9
	2	4	16
	2	5	25
4	2	2	4

Table 7: Selected Chebyquad test cases.

4.2.2 *Setting parameters*

Setting parameters for the step method

There are, for the step method, 3 parameters to be set: the initial threshold, the threshold step, and optionally the number of starting points.

The size of the step between 2 consecutive thresholds is critical. Indeed, it directly impacts both the computational effort and the accuracy of the results. A large step reduces the number of iterations, thus the CPU time. However it needs to be small enough to reach an acceptable accuracy of the solutions. A large step could lead to missing a particular level of Belief. If the optimal design is only associated with this missed level of Belief, it will not appear in the results at all.

Another parameter that can impact significantly the results is the number of starting points. A large number increases the chances of finding the global optimal results. But once again, the impact on the computational effort increases with it. Here again, different values will be considered.

Finally, the value of the initial threshold needs to be chosen. Remember that it should be large enough to ensure that a Belief of 1 is found. A simple way to do so is to compute the threshold giving a Belief of 1 for a given design, and use it as a starting point.

Table 8 summarises the settings for the step method in this work.

Parameter	**Value**
Initial threshold value	Value giving a Belief of 1 for a randomly selected design point (and used as initial starting point)
Step between 2 consecutive thresholds	.1, .05, .025, .0125, .00625
Number of initial design points (more than 1 is multi-start)	1, 2, 5, 10

Table 8: Setting parameters of the Step method.

Note that the local optimiser (here `fminsearch` from Matlab) could also be tuned. In this work, the default values are used.

Setting parameters for the bi-objective and multi-belief methods

The settings of both the bi-objective and multi-start methods are all related to the multi-objective global optimiser. NSGA2 is used here [Deb et al., 2002], and it needs 5 parameters to be set. These are:

POPULATION SIZE This is critical to the performance of the optimiser. A small population limits the number of function evaluations and increases the number of generations of evolution. However, a large population favours global search. NSGA2 usually performs best with a rather large population.

PROBABILITY OF CROSSOVER This controls the frequency at which two agents of the current population create two offspring by crossover. The higher this value, the more likely a crossover is to occur.

PROBABILITY OF MUTATION This controls the frequency at which an agent of the current population mutates to create an offspring. The higher this value, the more likely a mutation is to occur.

DISTRIBUTION INDEX FOR CROSSOVER This controls the spreading of the offspring of the current population. If a large value is chosen, the resulting offspring solutions are close to the parent solutions. On the other hand, for a small value, solutions away from parents are likely to be created [Deb et al., 2007, Deb and Agrawal, 1995].

DISTRIBUTION INDEX FOR MUTATION The distribution index of mutation works in the same way as for the crossover. The same comments are therefore applicable.

As for any test involving evolutionary algorithms, tuning the optimiser is tricky and can affect significantly the results. We set the probabilities and

distribution indices such that the convergence speed and the global exploration are balanced. The key parameter, however, is the size of the population. Therefore, 3 different sizes of population, adjusted to the number of design variables are tested for every problem.

Table 9 summarises the settings used here. The 3 population sizes are given separately in Table 10.

Parameter	**Value**
Probability of crossover	0.9
Probability of mutation	$1/(n_D + 1)$ *
Distribution index for crossover	10
Distribution index for mutation	25

*n_D is the number of design variables

Table 9: Setting parameters of NSGA2 for the bi-objective and multi-belief methods.

Design variables	**Population size**
1	16, 32 and 48
2	20, 40 and 60
3	24, 48 and 72
4	32, 64 and 96

Table 10: NSGA2 - Population sizes for the bi-objective and multi-belief methods.

For the multi-belief method only, the Belief levels tested could be selected. Once the combined BPA structure has been created, all the possible levels of Belief are known. One could potentially be interested in results with a Belief in a certain range, and therefore would select a subset of all possible Belief levels. Also, the number of objectives might become unmanageable for the global optimiser. Remember each level of Belief tested corresponds to 1 objective in the problem formulation. In this work, however, the complete set of possible Beliefs is used.

Setting parameters for the binary space partition method

The binary space partition (BSP) method has 2 settings only. The number of sample points and the step between 2 consecutive thresholds. The higher the number of sample points, the more information on the system function is available. Consequently, better results are expected. However, the time required to compute the approximation is linked to the number of sample points.

The same comments are applicable to the threshold step. The smaller it is, the greater the accuracy is expected to be. However, dividing by 2 the step means about twice as many steps will be necessary to solve the OUU problem. This is in fact analogous to the step method.

Table 11 summarises the settings used in this work.

Parameter	**Value**
Number of sample points	10^2, 10^3, $5 \cdot 10^3$, 10^4, $5 \cdot 10^4$, 10^5, $5 \cdot 10^5$, 10^6
Step between 2 consecutive thresholds	.1, .05, .025, .0125, .00625

Table 11: Setting parameters for the BSP method.

The way the sample points are selected can make a difference too. The sampling method can be chosen from a wide variety of techniques, and should be suited to the system function and the targeted goal of the sampling. Discussions about sampling can be found for example in [Swiler et al., 2006]. To illustrate the impact of the sampling, two different processes are tested. The first one is a classic, the Latin Hypercube [McKay et al., 1979]. An attempt at driving the sampling towards regions of interest has also been made. The idea proposed here is to focus the sampling on regions where the system function returns low values, i.e. regions of high performance of the system. For this, we use a global optimiser (NSGA2 in this work) applied to a pool function. The pool

function f_{pool} simply returns the value of the system function f if it is above a threshold ν, or the threshold itself otherwise.

$$f_{pool}(x,\nu) = \begin{cases} f(x) & \text{if } f(x) \geq \nu \\ \nu & \text{otherwise} \end{cases} \tag{4.7}$$

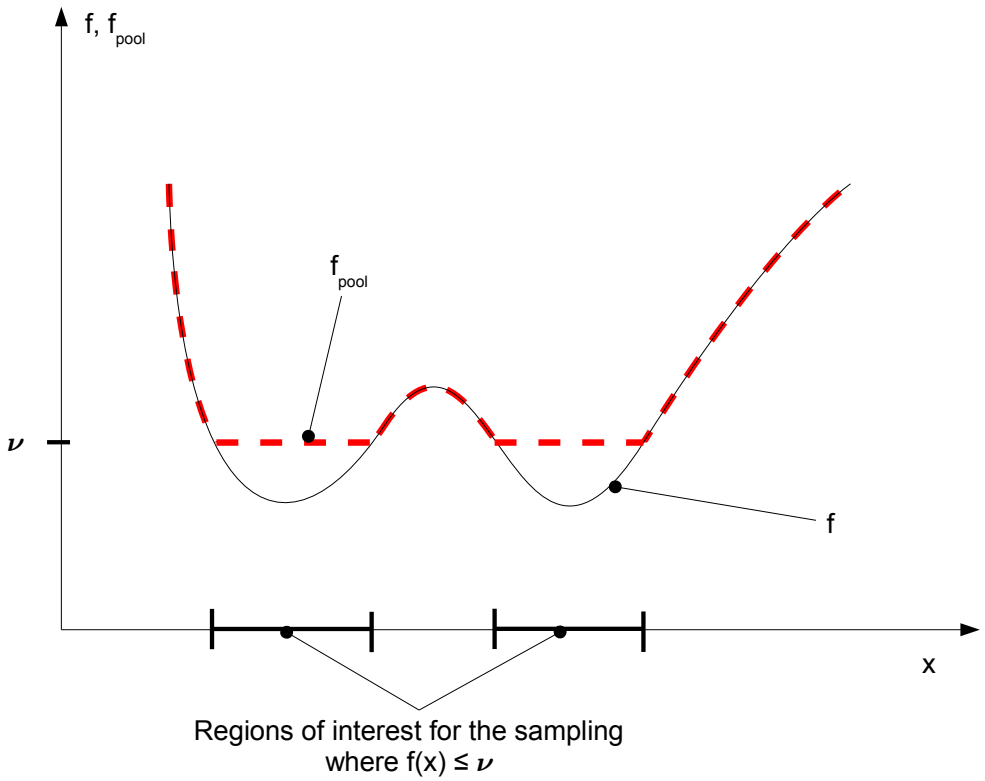


Figure 14: Example of a pool function.

Figure 14 illustrates a pool function. Because the minimum of the pool function is found for all points in any region of interest, the global optimiser will naturally focus its search on these regions, thus increasing more and more the sample density in these regions. Note that during the process, all points

evaluated by the global optimiser are saved as sample points, whether or not the system function returns a value below the threshold ν.

The value of the threshold needs to be large enough to ensure that a design region with a Belief of 1 is found. But the smaller the threshold is, the smaller the pool size, and thus the higher the density of sample point is expected to be. For this work, we set the threshold value to about 1.5 the optimal threshold giving a Belief of 1. Table 12 gives the value used for the different problems.

Design variables	**Uncertain parameters**	**Pool threshold**
1	1	0.5
	2	2.0
	3	3.5
	4	5.0
2	2	1.5
3	1	0.235
	2	1.0
	3	2.0
	4	3.0
4	2	0.8

Table 12: BSP method - Selected thresholds for the pool sampling.

The global optimiser needs also to be space. The aim here is to spread the sample points, but to have as many as possible within the pool, or nearby. Also, it is likely that more than one pool exists. The sampling is representative if all the pools are sampled. For this reason, higher values of the population size, and of the distribution indices for both crossover and mutation were chosen (c.f. Table 13).

Parameter	**Value**
Population size	100
Probability of crossover	0.9
Probability of mutation	$1/(n_D + n_U)$ *
Distribution index of crossover	20
Distribution index of mutation	40

*n_D is the number of design variables, n_U the number of uncertain parameters

Table 13: Setting parameters of NSGA2 for the pool sampling.

4.2.3 *Comparison metrics*

The performances of the proposed methods will be compared on the accuracy of the output results with respect to the optimal solutions identified in §4.1.3. The accuracy can be measured in the objective space (i.e. the threshold - Belief space in this work) or the search space (i.e. the design domain here). These are closely interdependent, but in cases where the problem has a flat optimum, or close local optima, differences can be seen. The metrics used to assess the accuracy are given in the next paragraph.

Being an approximation method, the BSP is expected to give less accurate results. However, it is designed to overcome the computational burden associated with the use of Evidence Theory. Therefore, the CPU time will be also considered.

Accuracy of the results

An optimisation method is primarily judged on the accuracy of the results (in the search as well as in objectives space) obtained for a given number of function evaluations. The metrics used in this chapter are defined below:

ERROR AREA The direct methods are exact, and thus can only find pessimistic Pareto fronts. However, the indirect methods may find over-optimistic results as well. To address this, the error area is used. It is defined as the area between the optimal Pareto front and the one found by each run. An example of this area is given in Figure 15. It is the shaded area between the optimal Pareto Front (continuous red curve) and the obtained results (dashed blue curve). As the threshold range can vary from one case to the other, the error area is normalised as follows:

$$A^*_{error} = \frac{A_{error}}{Y_{\underset{\max}{reqd}} - Y_{\underset{\min}{reqd}}} \tag{4.8}$$

where $Y_{\underset{\min}{reqd}}$ and $Y_{\underset{\max}{reqd}}$ are the minimum and maximum thresholds of the optimal Pareto front. In the remainder of this work, the "error area" refers to the normalised one.

DESIGN DISTANCE This metric measures the quality of the results in the search space, that is how far from the optimal designs the found ones are. To take into account the fact that a different design can be associated with each level of Belief, the design distance is defined as follows:

$$\rho_{\mathbf{d}} = \frac{1}{n_{bel}} \sum_{i=1}^{n_{bel}} \|\mathbf{d}_i - \mathbf{d}^*_i\| \tag{4.9}$$

In this equation, $\mathbf{d}^*_i$ is the optimal design for the i^{th} Belief level, and $\mathbf{d}_i$ the one found during the run. The factor $^1/_{n_{bel}}$ ensures this measure is relatively invariant with the number of Belief levels.

Different norms $\|\cdot\|$ can be used, but the classic Euclidean norm, L_2, has been preferred.

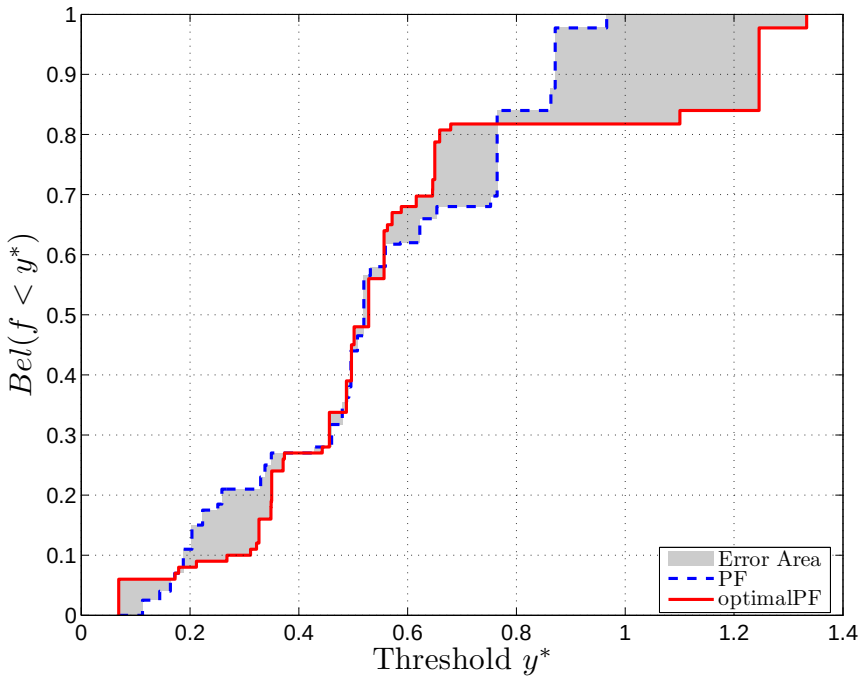


Figure 15: Definition of the error area (grey) between a Pareto front (dash blue) and the optimal Pareto front (continuous red).

CPU overhead for the indirect method

The sole purpose of the indirect method is to mitigate the computational effort. Therefore, it is of interest to measure the CPU time required to compute only the approximated Cumulative Belief Function $\widetilde{CBF}$. The variation of the CPU time with the number of designs variables, uncertain parameters and focal elements is crucial in assessing the performance of the indirect methods.

This concludes the presentation of the Chebyquad test campaign. The following three sections report the results. Comments and interpretations are provided throughout.

4.3 PERFORMANCE OF THE DIRECT METHODS

This section gives insights on the performance of the direct methods to solve the Chebyquad test cases. We will see that the anticipated weaknesses of the step method are confirmed. But firstly, the bi-objective and multi-belief methods are considered. The impact of the population size and the number of functions calls per design point are investigated. The quality of the results will be presented in detail in a separate section, and compared to the results obtained with the BSP.

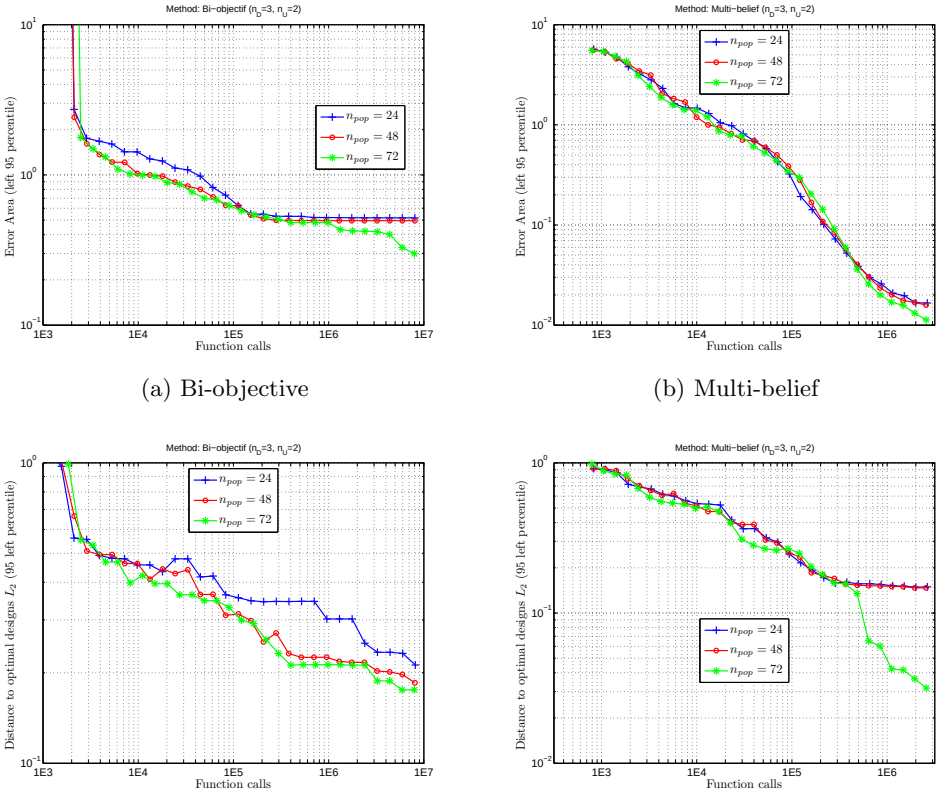
4.3.1 *Bi-objective and Multi-belief methods*

Impact of the population size

As described in §4.2.2, 3 population sizes have been considered for each case of the bi-objective and multi-belief methods.

Let's consider for instance the specific test of the Chebyquad with $n_D = 3$ and $n_U = 2$, and thus for the 3 tested population sizes. Figure 16a shows for the bi-objective method the 95 percentile of the error area varying with the number of function calls, while figure 16c gives the distance to the optimal designs. Figures 16b and 16d illustrate the same for the multi-belief method.

The bi-objective method appears sensitive to the population size. During the initial stages of the optimisation, a larger population gives lower values of the



(a) Bi-objective

(b) Multi-belief

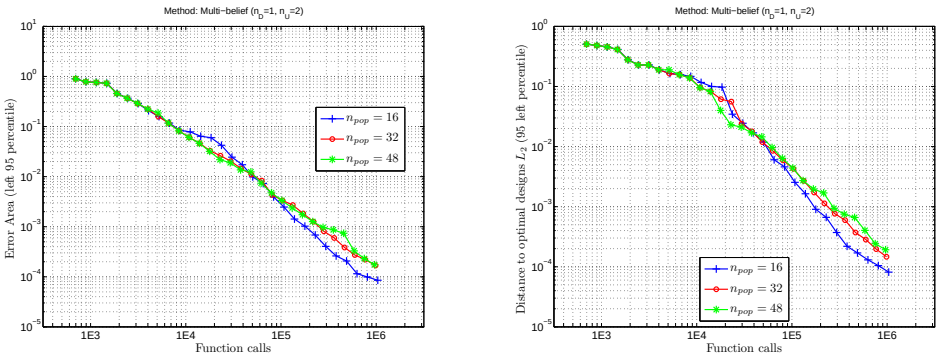
(c) Bi-objective

(d) Multi-belief

Figure 16: Bi-objective and Multi-belief - 95 left percentile of the error area (a)-(b) and distance to optimal designs (c)-(d), versus number of function calls, for the case $n_D = 3$ and $n_U = 2$.

error area. The optimiser requires less function evaluations to drive the agents towards the good regions of the design space. Additionally, a small population leads the optimiser to be stuck in a local optimum much earlier than with a larger population. Similar conclusions are applicable to the multi-belief case. However, it appears far less sensitive than the bi-objective case.

Most of the other test cases confirm this trend (c.f. Appendix B.1.1 and B.2.1 for the complete set of figures). However, the multi-belief method performs slightly better with the smallest population size than with the larger ones, on the Chebyquad cases with only 1 design variable. This is particularly noticeable on the test cases $n_D = 1$, $n_U = 2$, as illustrated in Figure 17.



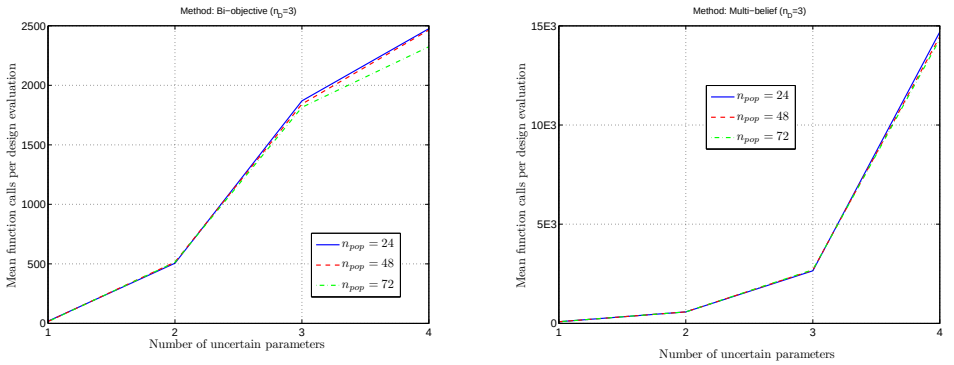
(a) (b)

Figure 17: Multi-belief - 95 left percentile of (a) the error area and (b) distance to optimal designs, versus number of function calls, for the case $n_D = 1$ and $n_U = 2$.

Number of function calls per design

In this subsection, the effect that the number of function calls per design case tested has on the peformance is investigated. Before going any further, it is worth recalling that, in the bi-objective method, a design point is defined as a design vector and a threshold, $(\mathbf{d}, y^*)$. This is not applicable to the multi-belief method, however, as the complete CBF curve is computed for each design.

The bi-objective method uses algorithmic tricks to limit as much as possible the number of function evaluations per design. Figure 18a shows that the average number of function calls per design evaluation on the $n_D = 3$ cases increases more or less linearly with the number of uncertain parameters. This demonstrates clearly the benefit of the algorithm introduced in §3.2.2.



(a) Bi-Objective (b) Multi-Belief

Figure 18: Mean number of function calls per design versus number of uncertain parameters on the $n_D = 3$ cases, with (a) the bi-objective and (b) the multi-belief methods.

For comparison, the same plot has been produced for the multi-belief method (c.f. Figure 18b). In this case, the number of function calls per design evaluation increases, as expected, exponentially with the number of uncertain parameters. Indeed, for each design evaluation, the CBF is computed. Thus every focal element needs to be considered. As the number of focal elements increases exponentially with the number of uncertain parameters, so does the number of function calls.

Now, let's consider a single simulation of the bi-objective method. The run #8 for the case $n_D = 3$, $n_U = 2$, and 24 agents in the population has been chosen. Figure 19 presents the variation of the total number of function calls as the number of design evaluations increases. The dashed line in the plot represents an hypothetical linear relationship, while the continuous line is the

actual simulation record. Up to about 4,500 designs tested, the number of function calls increases linearly with the number of design evaluations. After that, however, the additional design evaluations require fewer function calls. This characterises a convergence to a local optimum. Indeed, more and more design evaluations are prematurely terminated because they are found not to improve the current Pareto front. This is especially clear around 8,000 design evaluations. Similar behaviour is visible on any other runs and cases, but it is more or less marked and may occur later or sooner in the run.

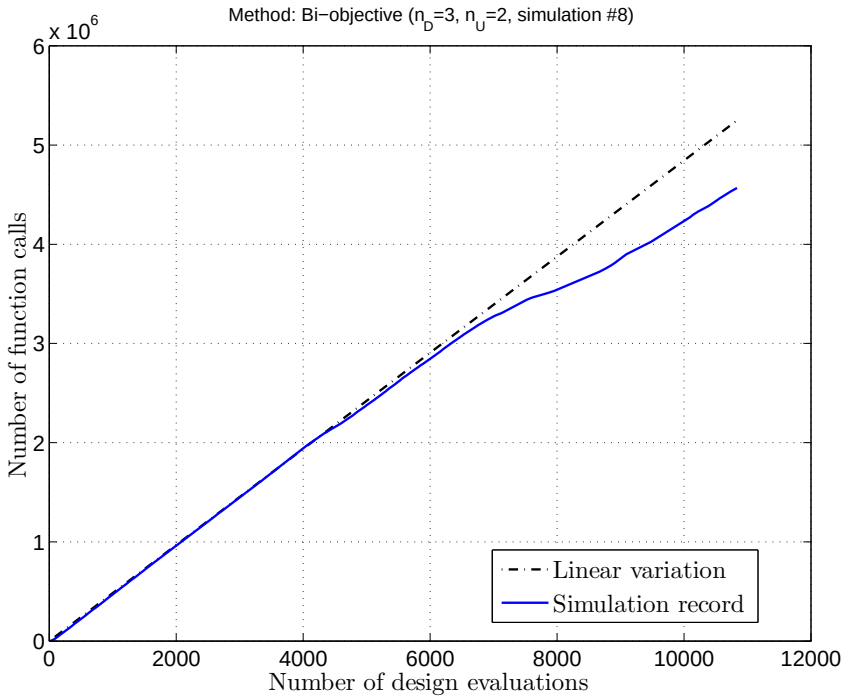


Figure 19: Number of function calls versus number of design evaluations for the bi-objective method - Case $n_D = 3$, $n_U = 2$, simulation #8 case.

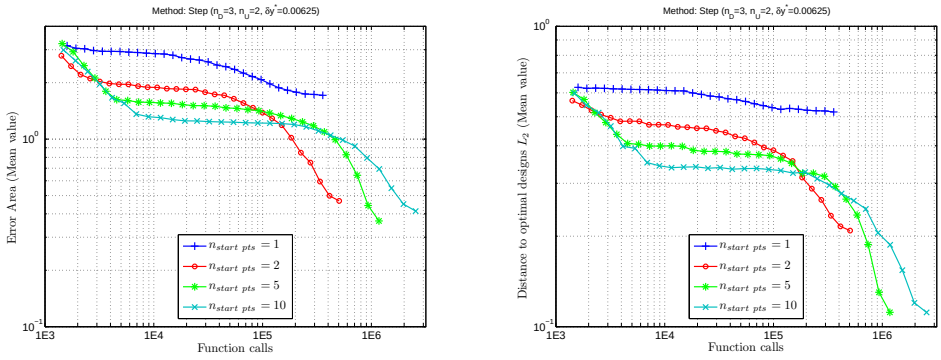
4.3.2 *Step method*

The results obtained with the step method are presented here. The potential issues with this method, foreseen in the previous chapter (c.f. §3.2.1), were verified on the Chebyquad test cases.

One problem envisaged was that the number of function calls cannot really be controlled. Simulations on the test cases with 3 uncertain parameters needed more than 100 millions function calls, that is more than 10 times what was run for the other two direct methods. For this reason, not all Chebyquad test cases have been run.

Also, the results were expected to be poor, due to the likelihood of the algorithm getting stuck in a local minimum. This can be seen for example in Figure 20, representing the case with 3 design variables and 2 uncertain parameters. The average error area (left) and distance to optimal designs (right) over the 100 runs are given during the run, as the number of functions calls increases. When only 1 starting point is used, the result is very poor, and there is no significant improvement with respect to the randomly chosen, initial design point. In particular, the fact that the distance to the optimal designs does not change much confirms that the agent remains close to the starting point. A closer look at individual runs that performed badly clearly showed this.

Using more than one starting point improves the performance. However, the results are still not as good as those returned by direct methods. Additionally, the number of function evaluations becomes very large. Remember that the complete Belief curve is known only at the end of the run. Stopping prematurely the iterations to control the number of function calls would truncate the part



(a) (b)

Figure 20: Step method - Average value over 100 runs of (a) error area and (b) distance to optimal designs, versus number of function calls, for the case $n_D = 3$ and $n_U = 2$.

with lower Belief. Therefore, the step method is too weak to be used on OUU problems, especially ones with multiple optimal designs.

4.4 PERFORMANCE OF THE BINARY SPACE PARTITION METHOD

This section gives insights in the performance of the Binary Space Partition method to solve the Chebyquad test cases. The impact of the number of sample points, threshold step and the sampling method are successively considered. Finally, special attention is given to the computational effort and its dependency on the above parameters.

4.4.1 *Impact of the number of sample points*

The indirect methods are based on the information gained via sampling the system function. The larger the set of sample points, the more accurate the obtained results should be. However, the purpose of any approximation is to

limit the number of sample points. This is particularly critical when the system function is computationally expensive.

Figure 21 is the box plot representing the size of the best design regions, for the case $n_D = 3$ and $n_U = 2$ when using pool sampling. 100 simulations have been run. It can be clearly seen that as the number of sample points increases, the mean size as well as the variance decreases. This illustrates that the returned results become more precise with a larger sample set.

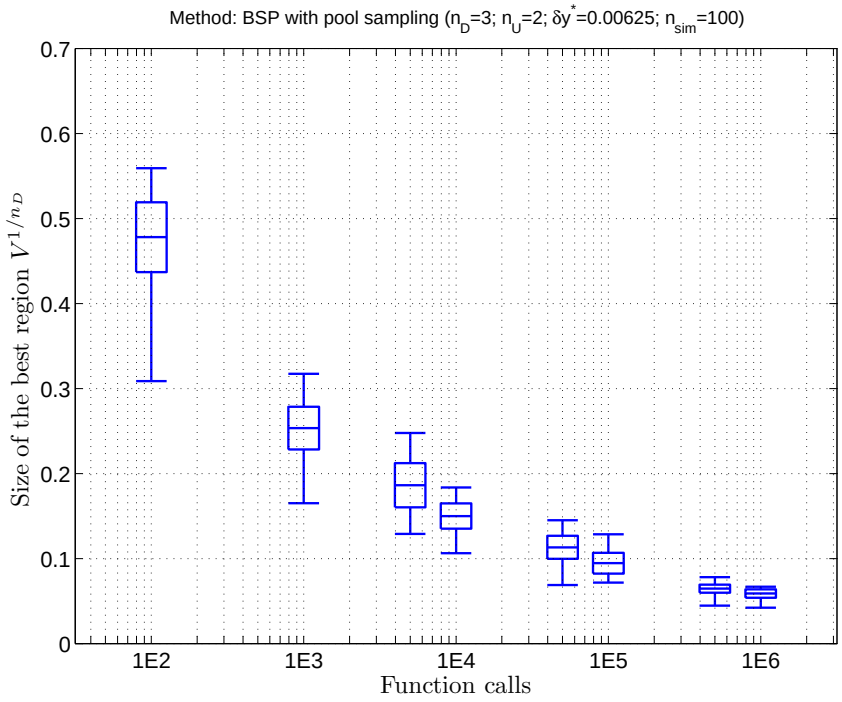
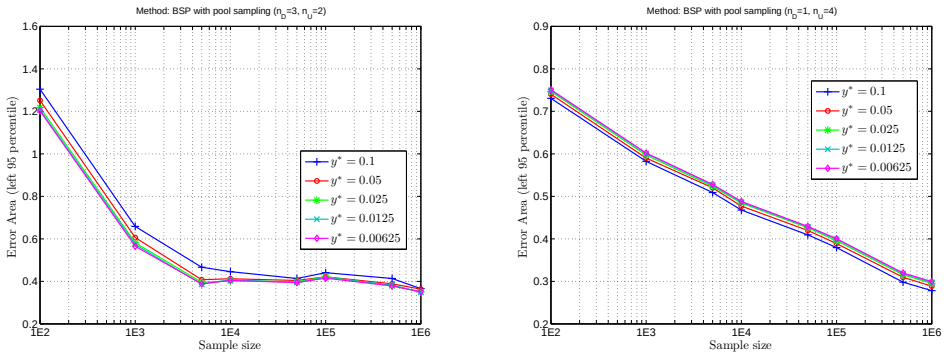


Figure 21: Size of the best design regions versus number of sample points for the BSP method - Pool sampling, case $n_D = 3$, $n_U = 2$.

Let's turn our attention to accuracy of the results. Figure 22a represents the variation of the error area for the same test case. The error area quickly decreases up to 5,000 sample points. Additional sample points do not significantly improve the error area: the method converged quickly on this example. Unsurprisingly, the more complex the problem is, the more sample points are required. As

a comparison, let us look at the case $n_D = 1$ and $n_U = 4$. Even though the number of dimensions is the same as the previous case (i.e. 5), the problem is more challenging as the number of focal elements increased from 4 to 16. Also, each of them is now a tesseract instead of a rectangle. It can be seen in Figure 22b that the error area does not reach a plateau, even with 1 million sample points.



(a) $n_D = 3$, $n_U = 2$ (b) $n_D = 1$, $n_U = 4$

Figure 22: Error area versus number of sample points for the BSP method - Pool sampling, cases with (a) $n_D = 3$, $n_U = 2$, and (b) $n_D = 1$, $n_U = 4$.

4.4.2 *Impact of the threshold step*

The threshold step directly influences the computational effort required to solve the problem and the accuracy of the results. The former is considered in §4.4.4 dedicated to CPU time, so only the latter is discussed here.

When using the proposed approximation method to solve an OUU problem, the Belief curve is in fact discretised. Indeed, the optimal Belief is computed for a finite set of thresholds. This inevitably introduces inaccuracy in the results. The smaller the threshold step, the more thresholds are tested, and thus the more accurate the results should be. This is hinted at in Figure 22a of the

previous section. A threshold step of 0.1 leads to less accurate results than smaller ones.

As the number of focal elements increases, the optimal Belief curve is composed of more and more steps (c.f. Figure 13). The influence of the threshold step on the accuracy of the results is clearer in the case with $n_D = 3, n_U = 2$, but with 25 focal elements instead of 4 as previously. This is illustrated in Figure 23.

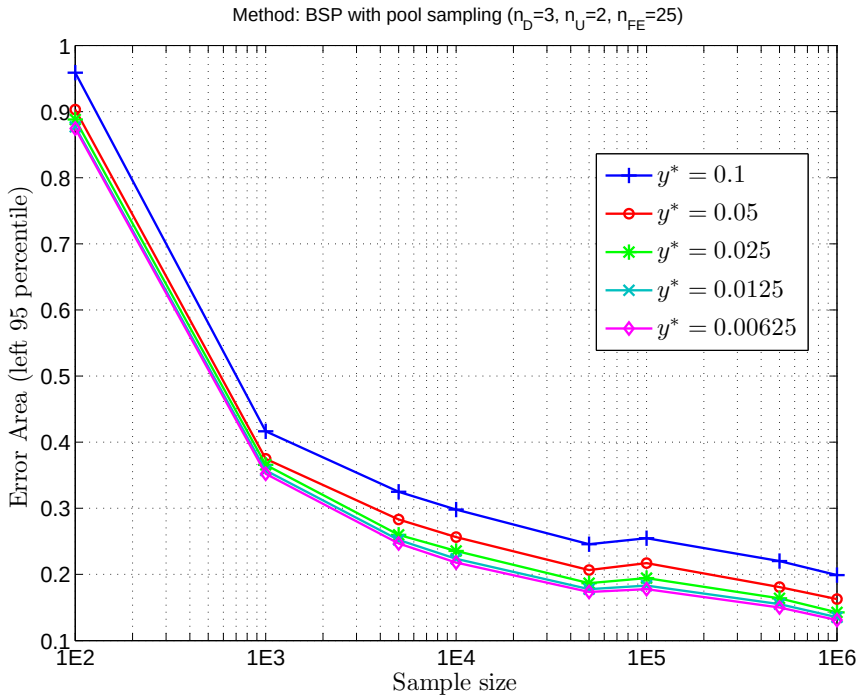


Figure 23: Error area versus number of sample points for the BSP method - Pool sampling, case $n_D = 3, n_U = 2, n_{FE} = 25$.

However, decreasing the threshold too much can quickly become ineffective, because the accuracy of the results is also very dependent on the sample used. See for example the case with $n_D = 1$ and $n_U = 4$ in Figure 22b of §4.4.1. In this case, the results are very similar, and in fact larger threshold steps gives slightly better results.

In light of the above, it is recommended to start the algorithm with a large threshold step, and reduce it until satisfying results are obtained. This way the threshold step would be small enough to capture the major variations of the optimal Belief curve, but not too small as to waste unnecessary CPU time.

4.4.3 *Impact of the sampling method*

The binary space partition has been associated with two different sampling techniques, uniform latin hypercube sampling (referred to as LHSU) and the pool method (referred to as pool sampling, or simply pool). In this section, the differences in performance between the two are assessed.

The LHSU is designed to sample uniformly the system function over its complete domain of definition. Pool sampling however aims at directing the sampling towards regions of interest. Thus, one can anticipate pool sampling will provide better results. The Chebyquad case with 3 design variables and 1 uncertain parameter clearly shows this (c.f. Figure 24). The difference is most significant for the smaller size of the sample set, and tends to reduce as convergence is approached.

Another comparison of the two sampling methods is done in Figure 25. Here, for 1 design variable and 3 or 4 uncertain parameters, the two sampling methods give similar results for small sample size. In fact, the LHSU slightly outperforms pool sampling. However, there is a critical sample size for which pool sampling visibly gives better results. The crossover occurs later in the $n_U = 4$ case than in the $n_U = 3$ case.

This behaviour can be explained as follows. For a small sample size, pool sampling returns points that are not spread widely enough to be representative of the system function. The influence of the initial population, composed of only

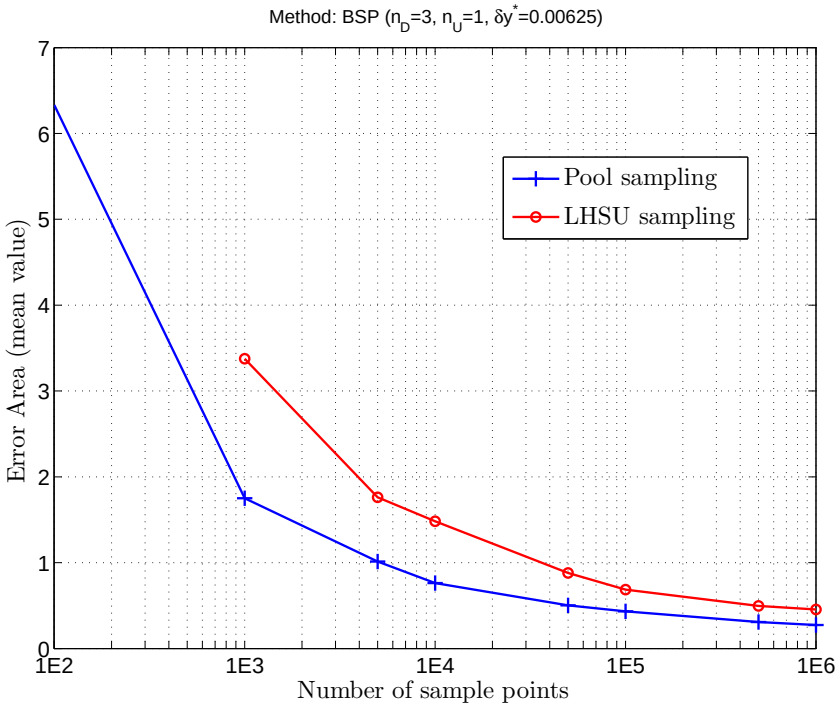
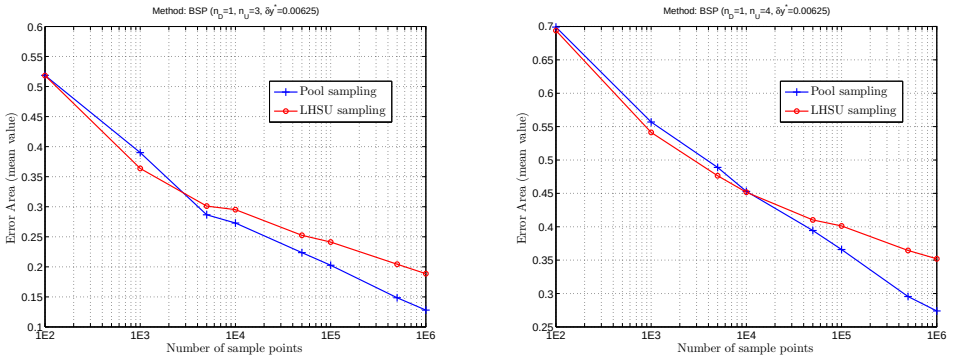


Figure 24: Comparison of the sampling methods for the BSP - Error area versus number of sample points for the case $n_D = 3, n_U = 1$.

100 individuals, remains very strong. The small number of generations did not allow the sample set to cover the pools. The subsequent sample points become more and more valuable as the genetic algorithm drives them towards the pools. On the other hand, the sample set given by the LHSU is by design well spread. Thus, as the number of sample points increases, a significant number fall into regions of little interest.

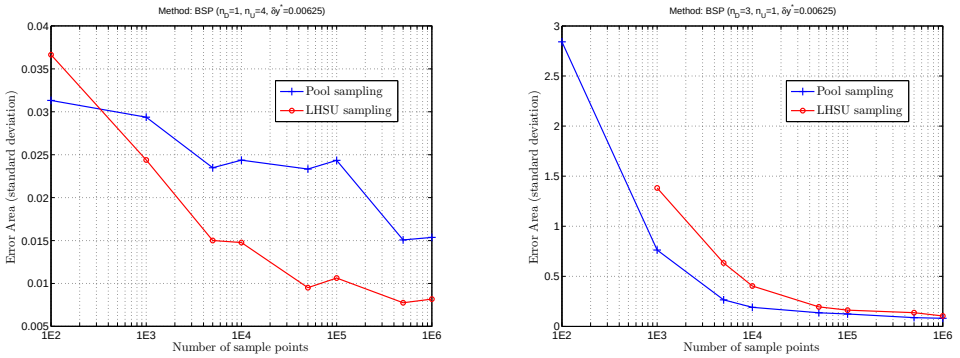


(a) $n_D = 1, n_U = 3$ (b) $n_D = 1, n_U = 4$

Figure 25: Comparison of the sampling methods for the BSP - Error area versus number of sample points for the cases (a) $n_D = 1, n_U = 3$ and, (b) $n_D = 1, n_U = 4$.

Finally, let's look at the variance of the results. On almost all the cases, the standard deviation of the error area is very similar for both sampling techniques, and small with respect to the mean value (see all the box plots of the error area in Appendix B.3.1). However, 2 cases stood out, namely the problem with $n_D = 1, n_U = 4$ and then $n_D = 3, n_U = 1$. Figure 26 shows for both cases the standard deviation of the error area over 100 simulations. In the first test case, the LHSU gives results with less variance, even if only marginally. But in the second case, it is the inverse, and pool sampling gives significantly more consistent solutions. This can be explained by considering the characteristics of both OUU problems. With 3 design variables and 1 uncertain parameter, the Chebyquad presents a very steep Pareto front (c.f. Figure 12). In other

words, the range of thresholds that is relevant is very narrow. Therefore it is vital that the sample points are located in regions where the system function's value is within, or close to, this range. The benefit of using pool sampling is evident in these circumstances. The case with 1 design variable and 4 uncertain parameters is quite different in that respect as the Pareto front is much flatter (c.f. Figure 11). There is not a lot of benefit in using pool sampling here, and, in fact, the randomness due to the global optimiser becomes visible.



(a) $n_D = 1, n_U = 4$ (b) $n_D = 3, n_U = 1$

Figure 26: Comparison of the sampling methods for the BSP - Standard deviation of the error area versus number of sample points for the cases (a) $n_D = 1, n_U = 4$ and, (b) $n_D = 3, n_U = 1$.

To conclude, pool sampling appears to be an improvement with respect to the LHSU. However, this improvement is most of the time modest. Problems that will benefit most from pool sampling are ones with large variation around the optimal designs, and/or a small range of thresholds of the optimal Belief curve. A few additional remarks can be made:

- The LHSU does not require any tuning, thus it is straightforward to use.
- Pool sampling requires a genetic algorithm (or similar) to drive the sampling process. This has three drawbacks: (i) there is an additional

computational cost, however small, associated with it, (ii) a tuning of the optimiser is required and (iii) the user needs to define the pool threshold. The latter is the most problematic, as this calls for some knowledge of the problem before tackling it. A badly selected pool threshold would compromise the quality of the sampling.

- Pool sampling makes it easy to adjust the sample size. One can generate an initial sample, apply the BSP to get a feeling for the OUU problem, and then restart the sampling process where it stopped, and so forth.

4.4.4 *CPU time*

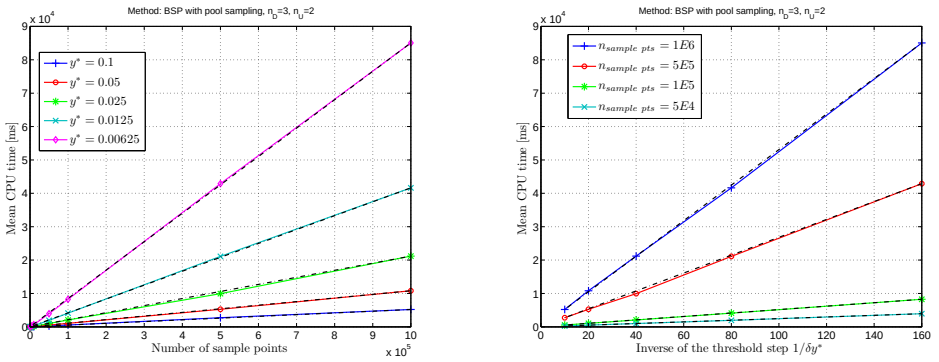
This section deals with the computational cost associated with the binary space partition method. The cost of the sampling is not considered here. Indeed, this mostly depends on the system function, whose computational cost can vary dramatically in real world problems.

The simulations have run on a Unix system, with a AMD Opteron™ dual core processor 275 (clock frequency: 2.2 GHz, cache: 1 MB, RAM: 4 GB). The BSP code is written in C. The actual values of the CPU times presented here are of little interest, as they depend on the machine used. However, the order of magnitude and more importantly the dependency on the OUU problem and the settings is relevant. The threshold step, sample size, dimensionality of the problem, number of focal elements are considered below one after another.

The threshold step has a very predictable influence. Indeed, it directly influences the number of iterations that will be performed, each iteration being composed of (i) partitioning the space and (ii) finding the best approximated Belief. As the number of iterations is inversely proportional to the threshold

step, so is the CPU time. Figure 27b illustrates this on the Chebyquad case with $n_D = 3$ and $n_U = 2$.

The number of sample points impacts linearly on the binary space partition. This is because the function value at each and every point needs to be compared to the threshold. Additionally, the more sample points, the more partitions are created. Thus the Belief approximation is also more computationally expensive. Figure 27a shows, for the case $n_D = 3$, $n_U = 2$, the impact of the sample points on the CPU time required to compute the approximated Belief curve. For any threshold step, the CPU time is linear. This is general to all cases, but the gradient may vary significantly.



(a) (b)

Figure 27: CPU time required to compute the approximated Belief curve versus (a) the number of sample points and, (b) the inverse of the threshold step - Pool sampling, case $n_D = 3, n_U = 2$ (the dash lines mark a linear behaviour)

Similarly, the dimensionality of the problem should have a linear impact on the CPU time required to partition the domain. This is illustrated in Figure 28, which gives the mean CPU time as a function of the number of design variables. The threshold step is fixed, and the 4 curves are given for 4 different sample sizes. Also, the number of uncertain parameters is kept constant, thus so is the number of focal elements. This means the CPU time required to compute

the Belief approximation depends in this case only on the number of design variables. The same applies to the CPU time required to partition the space.

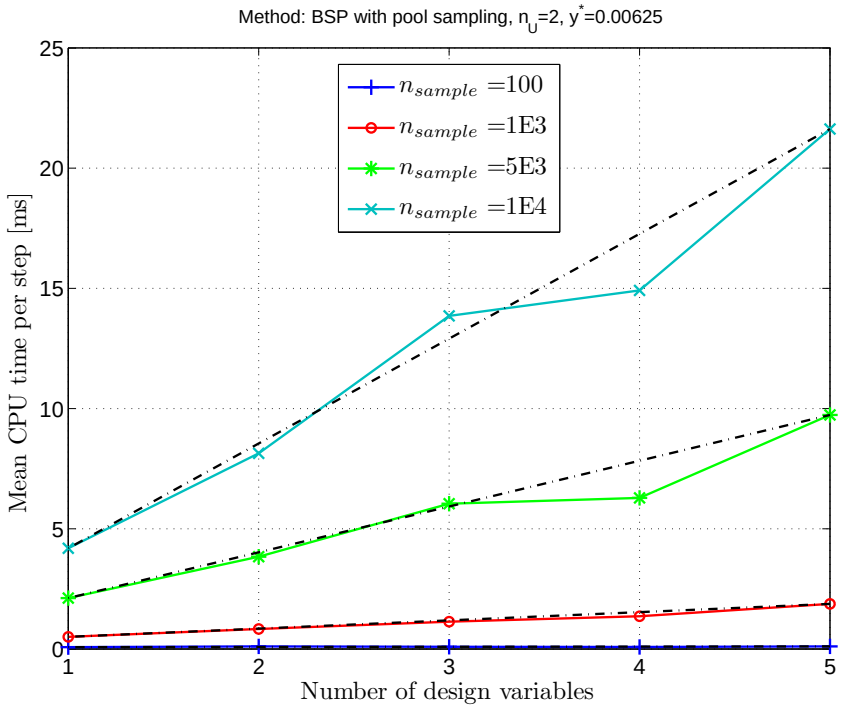


Figure 28: CPU time required to compute the approximated Belief curve versus the number of design variables - Pool sampling, case $n_U = 2$, $y^* = 0.0065$ (the dash lines mark a linear behaviour).

Based on the algorithm described in §3.3.3, the CPU time required to compute the approximated Belief should increase linearly with the number of focal elements of the problem at hand. Indeed, every focal element must be tested to access the Belief value. Note that, as explained earlier, the total number of focal elements increases exponentially with the number of focal elements of each individual uncertain parameter (c.f. equation (2.5)). Figure 29 shows the mean CPU time of the BSP approximation with constant numbers of sample points, design variables and uncertain parameters. Therefore, the CPU time associated with the partition phase is fixed. The variation of CPU time is here solely due to computing the Belief approximation. As expected, the

increase of CPU time appears on the plot linear with the total number of focal elements. Importantly, the increase is very flat, suggesting a weak dependence of the BSP performance on the number of focal elements.

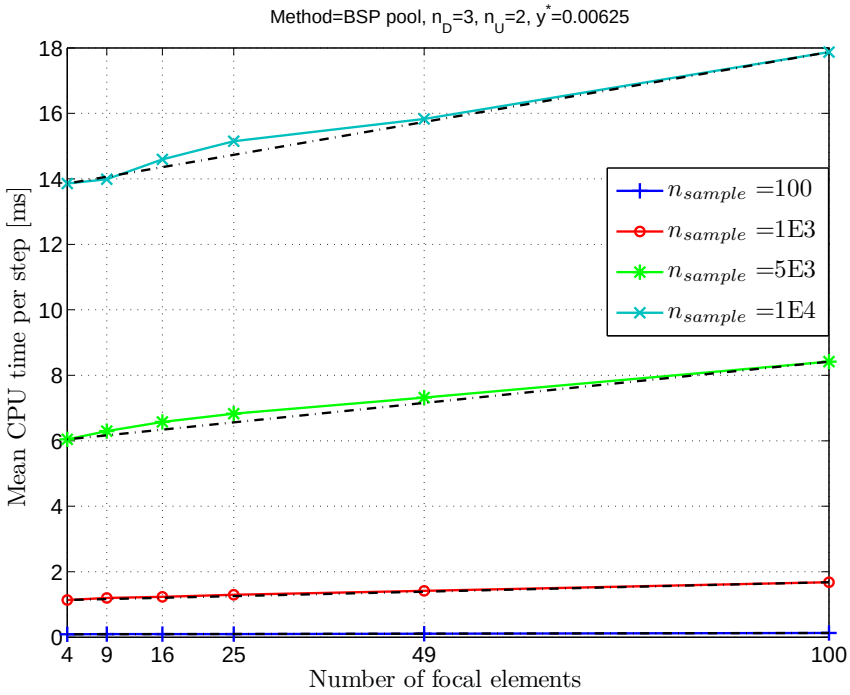


Figure 29: CPU time required to compute the approximated Belief curve versus the number of sample points - Pool sampling, case $n_D = 3, n_U = 2$.

We have shown here that, as expected, the CPU time of the BSP varies:

- linearly with the number of dimensions of the problem to partition the space.
- linearly with the number of sample points to partition the space.
- linearly with the threshold step.
- exponentially with the number of focal elements of each uncertain parameter when calculating the approximated Belief. Additionally, the results

suggest a weak dependence of the performance on the number of focal elements.

The main purpose of the indirect method is to control the CPU time required to solve an OUU problem. Direct methods face the issue of exponential increase of CPU time with the number of focal elements of each uncertain parameter. The proposed binary space partition method confines the exponential increase to the smallest part possible of the algorithm, the computation of the Belief approximation. This is directly linked to the nature of the exponential increase of the number of focal elements with the number of uncertain parameters. The Chebyquad tests have illustrated that this exponential variation is in fact very flat. Thus the BSP method appears to handle problems with very large sets of focal elements in a matter of seconds. This should be tested again on the problems related to space mission design in the next chapter.

4.5 QUALITY OF THE RESULTS

Finally, the quality of the results given by each method is compared. The aim is to provide insights into the strengths and weakness by comparing on all Chebyquad test cases firstly the accuracy of the Belief curve, and secondly how far the returned designs are from the optimal ones. Ultimately, this would help the end user to select the appropriate method to solve practical OUU problems.

4.5.1 *Accuracy in the objective domain*

On all the plots of this subsection, the error area as defined in §4.2.3 versus the number of function calls is given. The results of the bi-objective and the

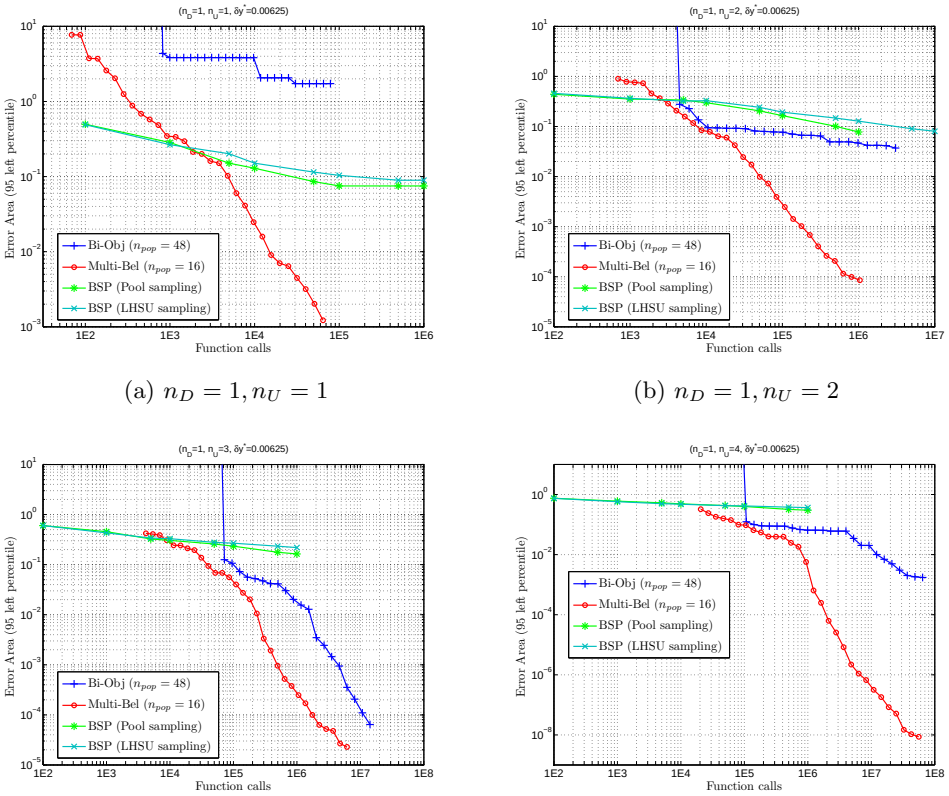
multi-belief methods for the direct cases, and the BSP with both sampling methods for the indirect cases are provided, on the same graph, for direct comparison. Moreover, the plots are grouped into sets of 4 as follows:

- Figure 30 for the 4 test cases with 1 design variable,
- Figure 31 for the 4 test cases with 3 design variables,
- Figure 32 for the 4 test cases with 2 uncertain parameters,
- Figure 33 for the 4 test cases with 3 design variables, 2 uncertain parameters and varying numbers of focal elements.

The results obtained with the smallest threshold step are presented for the indirect methods, as this provided the best results. For the direct methods, the results obtained with the largest population size are given for the same reason. The only exception is with the multi-belief method when applied to the test cases with 1 design variable, where the smallest population size is used (c.f. §4.3.1).

Based on these figures, the following comments can be made:

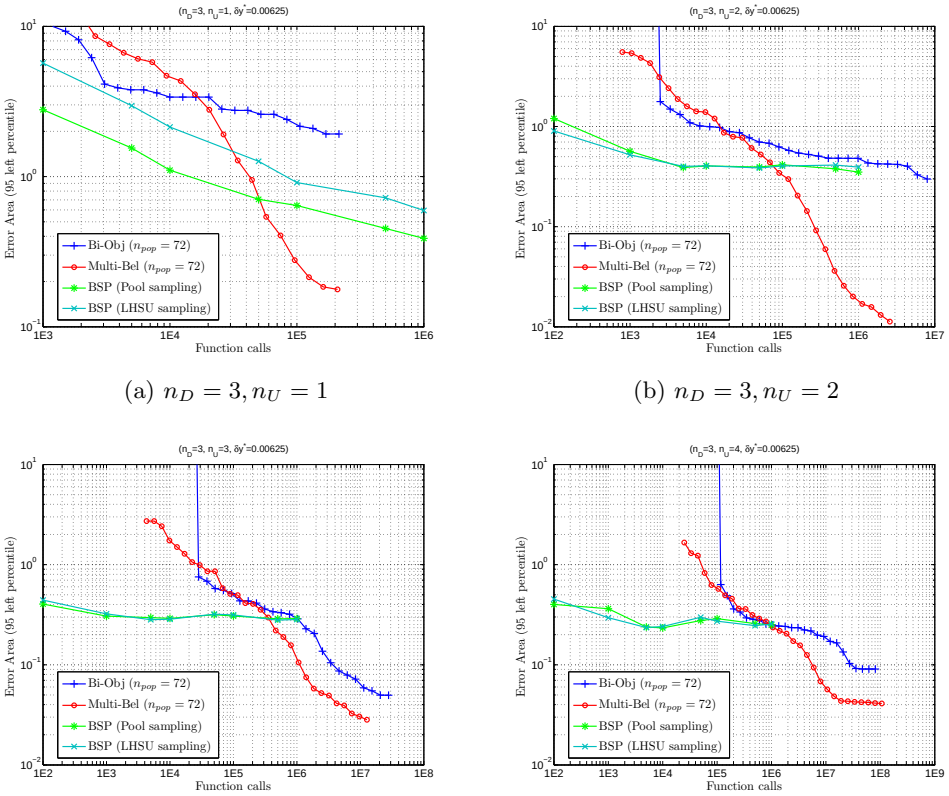
- The multi-belief method clearly dominates the bi-objective one on the test cases with 1 design variable. As the number of design variables increases, the bi-objective method performs in comparison better, especially for small number of function calls. The same remark is valid when the number of focal elements increases. In fact, the test cases with 16 and 25 focal elements are the only ones for which the bi-objective method outperforms the multi-belief one for most, if not all, numbers of function calls, even if only marginally.
- The approximated Belief curve obtained with the BSP is better for small numbers of function calls on all test cases. The point at which



(a) $n_D = 1, n_U = 1$ (b) $n_D = 1, n_U = 2$

(c) $n_D = 1, n_U = 3$ (d) $n_D = 1, n_U = 4$

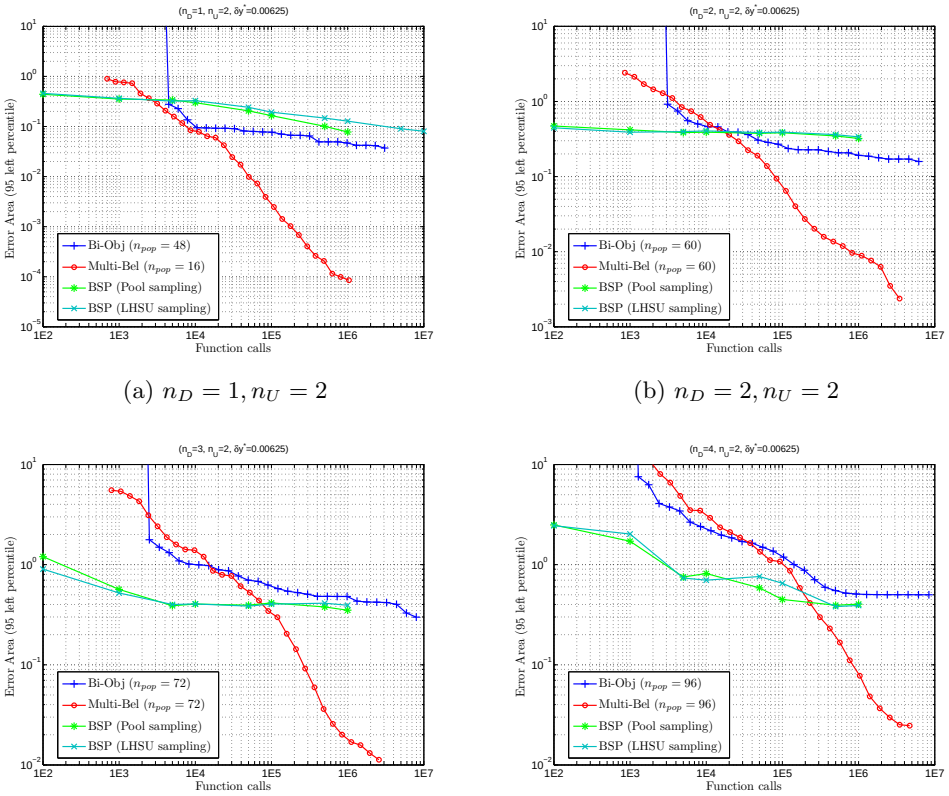
Figure 30: Comparison of all the methods - Error area (95%) versus number of function calls for the cases with $n_D = 1$ and (a) $n_U = 1$, (b) $n_U = 2$, (c) $n_U = 3$, and (d) $n_U = 4$.



(a) $n_D = 3, n_U = 1$ (b) $n_D = 3, n_U = 2$

(c) $n_D = 3, n_U = 3$ (d) $n_D = 3, n_U = 4$

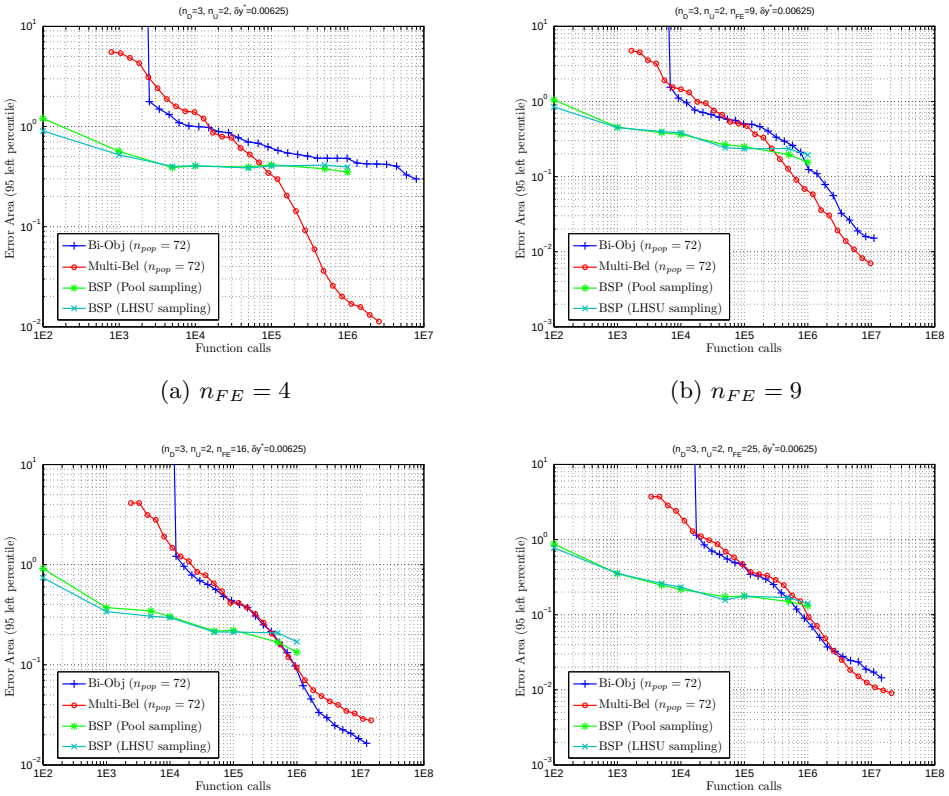
Figure 31: Comparison of all the methods - Error area (95%) versus number of function calls for the cases with $n_D = 3$ and (a) $n_U = 1$, (b) $n_U = 2$, (c) $n_U = 3$, and (d) $n_U = 4$.



(a) $n_D = 1, n_U = 2$ (b) $n_D = 2, n_U = 2$

(c) $n_D = 3, n_U = 2$ (d) $n_D = 4, n_U = 2$

Figure 32: Comparison of all the methods - Error area (95%) versus number of function calls for the cases with $n_U = 2$ and (a) $n_D = 1$, (b) $n_D = 2$, (c) $n_D = 3$, and (d) $n_D = 4$.



(a) $n_{FE} = 4$ (b) $n_{FE} = 9$

(c) $n_{FE} = 16$ (d) $n_{FE} = 25$

Figure 33: Comparison of all the methods - Error area (95%) versus number of function calls for the cases with $n_D = 3, n_U = 2$ and (a) $n_{FE} = 4$, (b) $n_{FE} = 9$, (c) $n_{FE} = 16$, and (d) $n_{FE} = 25$.

the direct methods perform better changes with the complexity of the problem. With 1 design variable, the crossover occurs between 1 to 10×10^3 function calls. With 3 design variables, this occurs much later, around 50 to 1000×10^3. In any case, the crossover occurs later as the number of uncertain parameters or focal elements increases.

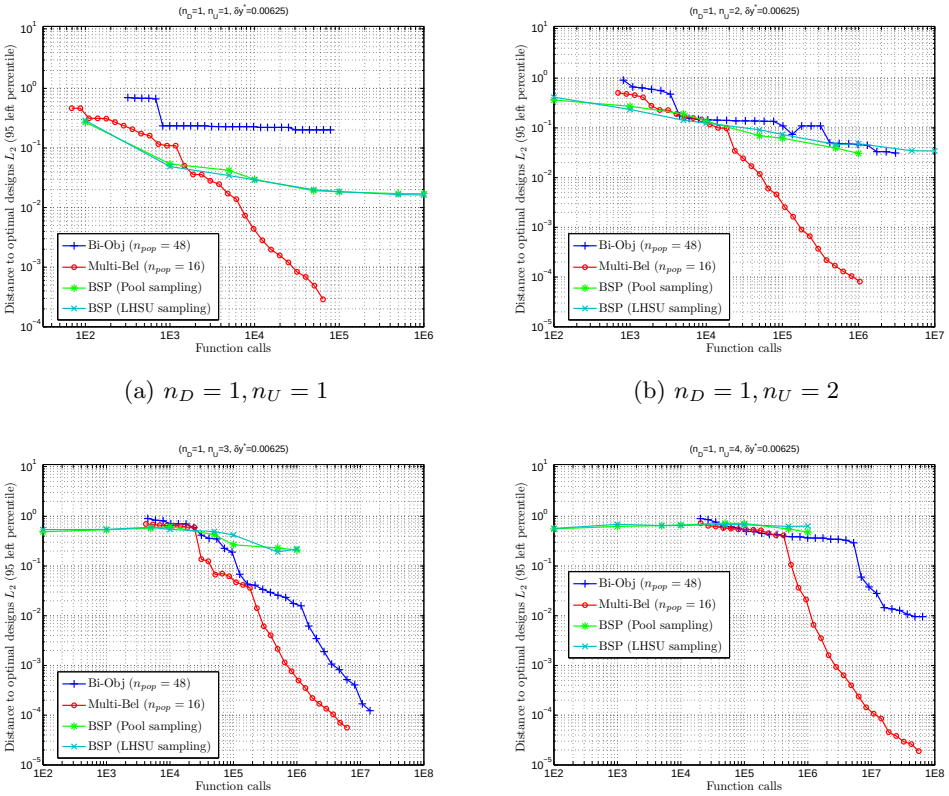
- The bi-objective method performs particularly poorly in the test cases with 1 design variable and 1 uncertain parameter.
- The BSP displays an oscillating behaviour for the case $n_D = 3, n_U = 4$ (and also, although barely noticeably for $n_D = 3, n_U = 3$). The additional sample points do not improve the results, but rather drive the algorithm to identify at best distinct regions. This could be due to very close local optima, hard to distinguish with an approximation technique.

4.5.2 *Accuracy in the design domain*

Let's turn our attention now to the performance with respect to the best designs found. The plots are again grouped into sets of 4 as follows:

- Figure 34 for the 4 test cases with 1 design variable,
- Figure 35 for the 4 test cases with 3 design variables,
- Figure 36 for the 4 test cases with 2 uncertain parameters,
- Figure 37 for the 4 test cases with 3 design variables, 2 uncertain parameters and various numbers of focal elements.

The comments made above remain valid when considering the distance to the optimal design points. One can note that the BSP tends to perform comparatively better in the search space than in the objective space. In the case



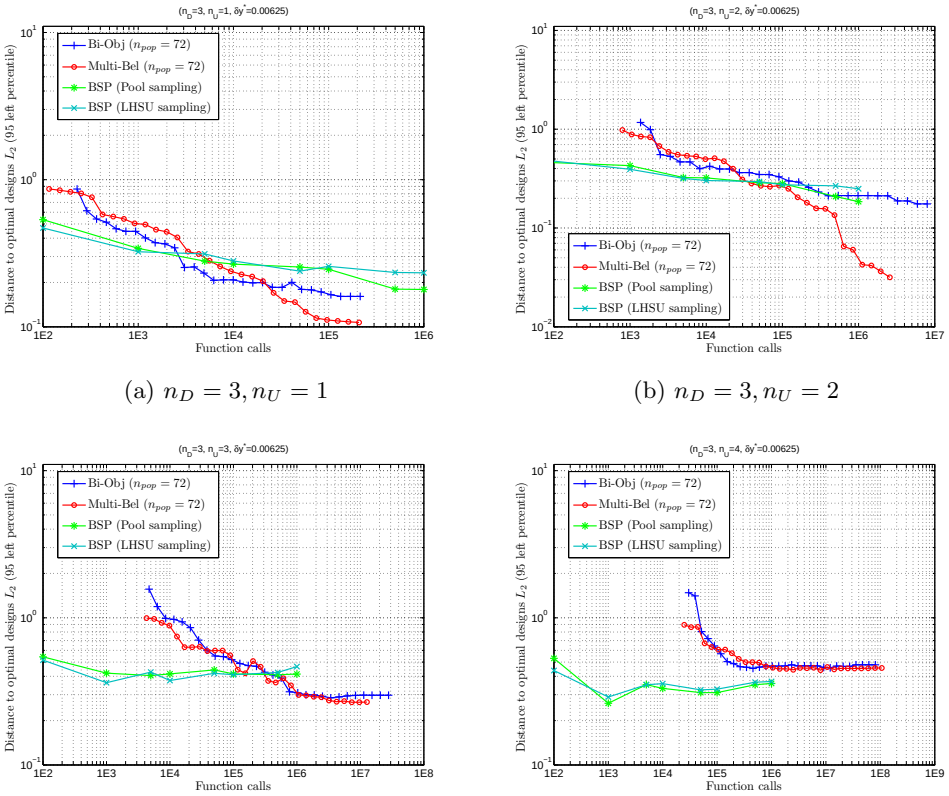
(a) $n_D = 1, n_U = 1$

(b) $n_D = 1, n_U = 2$

(c) $n_D = 1, n_U = 3$

(d) $n_D = 1, n_U = 4$

Figure 34: Comparison of all the methods - Distance to optimal designs (95%) versus number of function calls for the cases with $n_D = 1$ and (a) $n_U = 1$, (b) $n_U = 2$, (c) $n_U = 3$, and (d) $n_U = 4$.



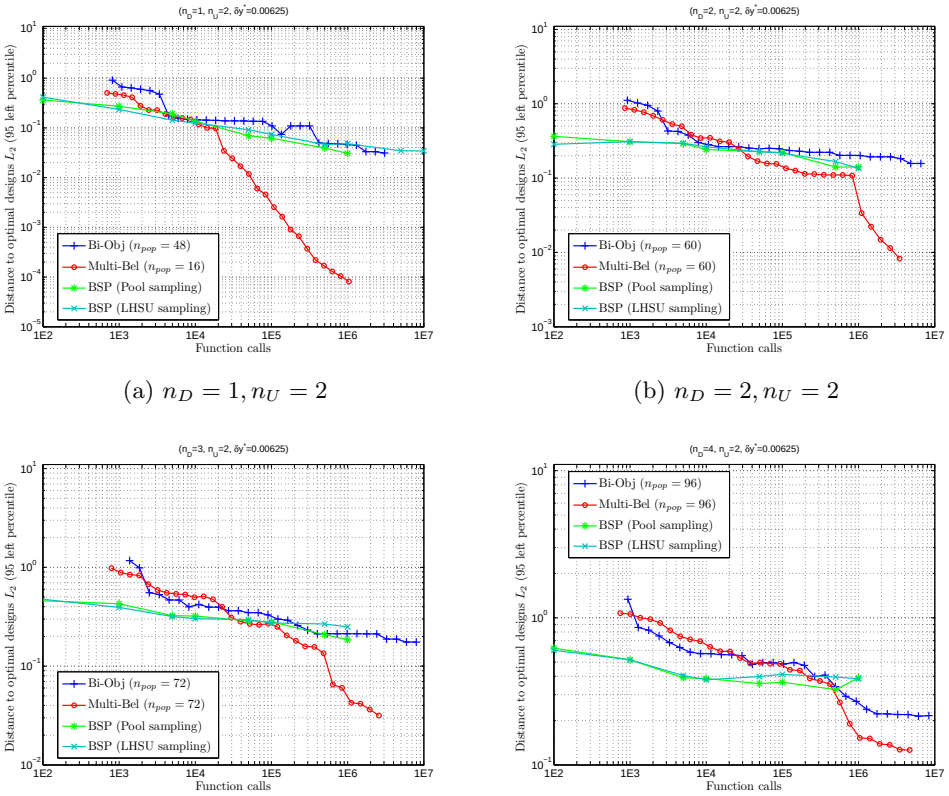
(a) $n_D = 3, n_U = 1$

(b) $n_D = 3, n_U = 2$

(c) $n_D = 3, n_U = 3$

(d) $n_D = 3, n_U = 4$

Figure 35: Comparison of all the methods - Distance to optimal designs (95%) versus number of function calls for the cases with $n_D = 3$ and (a) $n_U = 1$, (b) $n_U = 2$, (c) $n_U = 3$, and (d) $n_U = 4$.



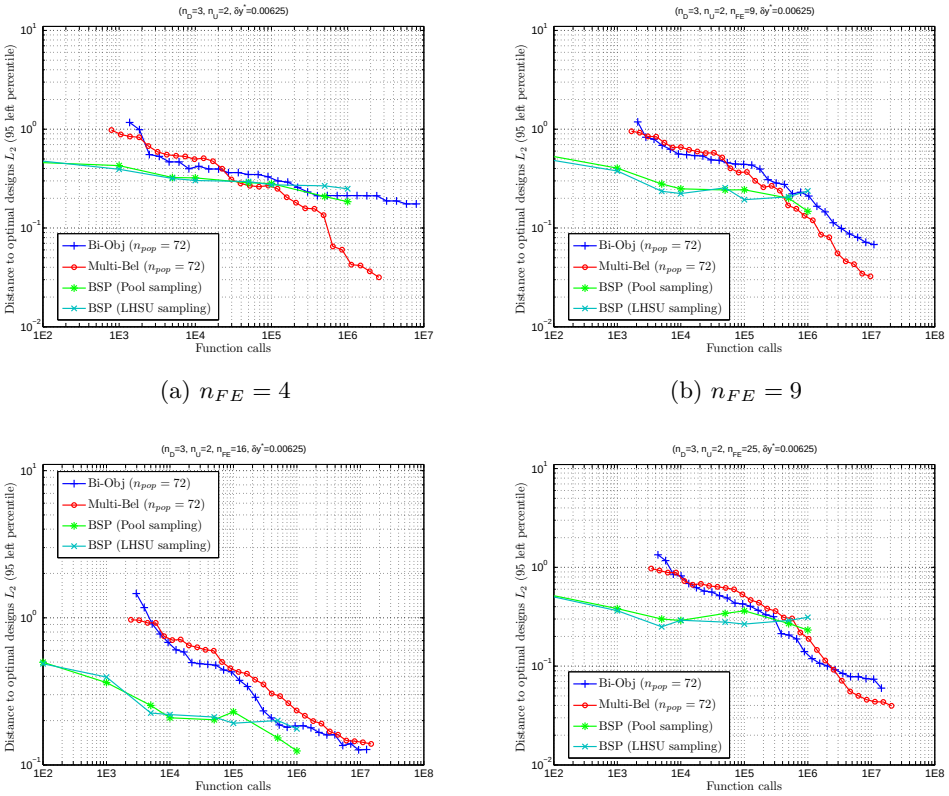
(a) $n_D = 1, n_U = 2$

(b) $n_D = 2, n_U = 2$

(c) $n_D = 3, n_U = 2$

(d) $n_D = 4, n_U = 2$

Figure 36: Comparison of all the methods - Distance to optimal designs (95%) versus number of function calls for the cases with $n_U = 2$ and (a) $n_D = 1$, (b) $n_D = 2$, (c) $n_D = 3$, and (d) $n_D = 4$.



(a) $n_{FE} = 4$ (b) $n_{FE} = 9$

(c) $n_{FE} = 16$ (d) $n_{FE} = 25$

Figure 37: Comparison of all the methods - Distance to optimal designs (95%) versus number of function calls for the cases with $n_D = 3, n_U = 2$ and (a) $n_{FE} = 4$, (b) $n_{FE} = 9$, (c) $n_{FE} = 16$, and (d) $n_{FE} = 25$.

with 3 design variables and 4 uncertain parameters, the best design points are closer to the optimal ones than those found by either direct method. However, in this test, the results obtained by the direct methods are fairly far away. This could be explained by (i) the existence of similar local optima and/or (ii) the dimensionality of the problem. With a total of 7 variables (the most tested in this work), it is likely that more function calls are necessary to converge closer to the set of optimal designs.

4.6 CONCLUSIONS

The extensive test campaign presented in this chapter was carried on the analytical Chebyquad problems. It provided useful information on the different proposed methods to solve OUU problem in the frame of Evidence Theory.

Unsurprisingly, the step method is too weak to solve even fairly simple OUU problems. It fails especially to identify distinct optimal designs at different levels of Belief.

Of the two other direct methods, the multi-belief method has a better overall performance. This is particularly noticeable in problems with small numbers of design variables and focal elements. As the complexity of the problem increases, the bi-objective method becomes more and more valuable, especially if the number of function calls is to be limited. This is due to the algorithmic tricks introduced in this work, which proved their effectiveness. Also, in contrast to the multi-belief method, the bi-objective formulation is fairly sensitive to population size. It appears preferable to use a rather large population to achieve best performance.

The BSP method proved to be a pertinent way to tackle OUU problems if the number of function calls is to be kept small. Once the sampling is

available, the computational time required to approximate the optimal Belief curve and identify good designs is very reasonable. It remains exponential with the number of focal elements, but this is confined to a very limited part of the algorithm, accounting for a fraction of the total in even the most complex Chebyquad test cases. The BSP method can handle problems with a very large set of focal elements in a matter of seconds.

Two sampling techniques have been associated with the BSP. Compared to Latin Hypercube Sampling, the proposed pool sampling method slightly improves the results. More importantly, it makes it easy to adjust the sample size by restarting the sampling if the BSP outputs are not good enough. However, the user needs to define the pool threshold which implies some knowledge of the problem before tackling it. A badly selected pool threshold would compromise the quality of the sampling.

The 3 methods are of interest, and can be selected depending on the problem at hand. General guidelines could be summarised as follows:

- If little is known about the problem, the BSP can be used to quickly extract the major trends, thus helping the analyst to select (and tune) the solver.

- If the system function is computationally expensive, the BSP is clearly a wise choice.

- For problems with large numbers of variables and/or focal elements, the bi-objective method should be considered instead of the multi-belief. This is also the case if only a part of the Belief curve is of interest. Indeed, the threshold is a variable to the genetic algorithm, and the analyst can constrain it and thus focus the search on Belief range of interest.

Improvements to the algorithms presented in this work are of course possible. Greater attention could be given to the sampling process associated with the indirect method. The pool function, for instance, could be enhanced by introducing an additional pool threshold. The idea is to further reduce the pool size, driving the sample points away from regions where the system function returns too high or too low values. A second modification would be to limit the number of partitions along the design dimensions. Because this is an approximation method, the precision of the designs cannot be very high, and therefore it does not make sense to have tiny regions of interest returned by the solver. This should increase significantly the probability that the optimal designs are contained in the best design regions identified by the BSP. A third modification that comes to mind concerns the bi-objective method. The threshold is treated by the optimiser in the same way as the designs variables. Thus crossovers between the threshold and one of the design variables occur. This is likely to impair the performance of the genetic algorithm. This issue could be resolved by preventing the genetic algorithm from considering, during the creation of a new generation, the variable "threshold" for crossover.

5

APPLICATIONS TO SPACE MISSION DESIGN

This chapter is dedicated to applying the various methods introduced in the previous chapters to solve space-related reliable design problems. Two different test cases are proposed, a BepiColombo-like mission to Mercury, and a feasibility study to rendez-vous with an asteroid.

These two applications are representative of exploration and science missions (as opposed to commercial applications). The reason for this choice is because risk is more accepted in space exploration missions, a reminder, and remainder, of the pioneer spirit of the early years of the space age. Additionally, science missions are more subject to uncertainty, especially at the preliminary stage, generally due to the limited knowledge of the operational environment. Indeed, the design of a science spacecraft is each time strongly tailored to the specifics of the mission, while commercial satellites are in comparison mass-produced. The consequence is that a lot of feasibility studies for space science missions are performed, but only a handful moves on to the implementation stage.

A second commonality of the test cases is the use of Solar Electric Propulsion (SEP). The rationale for this choice is due to the fact that such a propulsion system has a significant impact, in particular mass-wise, on the overall configuration of the spacecraft. The mass of propellant is significantly reduced with respect to a conventional chemical propulsion system, but the need for large solar arrays, more complex harness and power processing units, and even larger radiator, could negate the expected mass savings. Also, the substantially lower thrust levels available with SEP than impulsive propulsive systems impacts

greatly on the trajectory, and its optimisation is a difficult problem. Clearly, therefore, designing a spacecraft with SEP is a challenging multidisciplinary activity.

The first test case is largely based on the BepiColombo mission to Mercury. A comparison of the direct and indirect methods to solve the OUU problem is made. The conclusions are in line with what was observed in the previous chapter.

The second application is a feasibility study of a mission to the asteroid 1999JU3. The scenario is of a space mission analysis required to find a reliable preliminary design without initial knowledge of the problem, and limited computational power and time allocated for the task.

5.1 A CASE STUDY: THE BEPICOLOMBO MISSION

In this section, we will present the results obtained when the direct and indirect methods, introduced and discussed in the previous chapters, are applied to the preliminary design of a BepiColombo-like mission. The objective of the design is to minimise the mass of the subsystems linked to the low-thrust propulsion. We assume that the design is at a preliminary stage, and therefore consider uncertainties on a few parameters.

After introducing the actual BepiColombo mission, the mass modelling of the subsystems is detailed and the OUU problem is formulated. Results and comments conclude the presentation of this first test case.

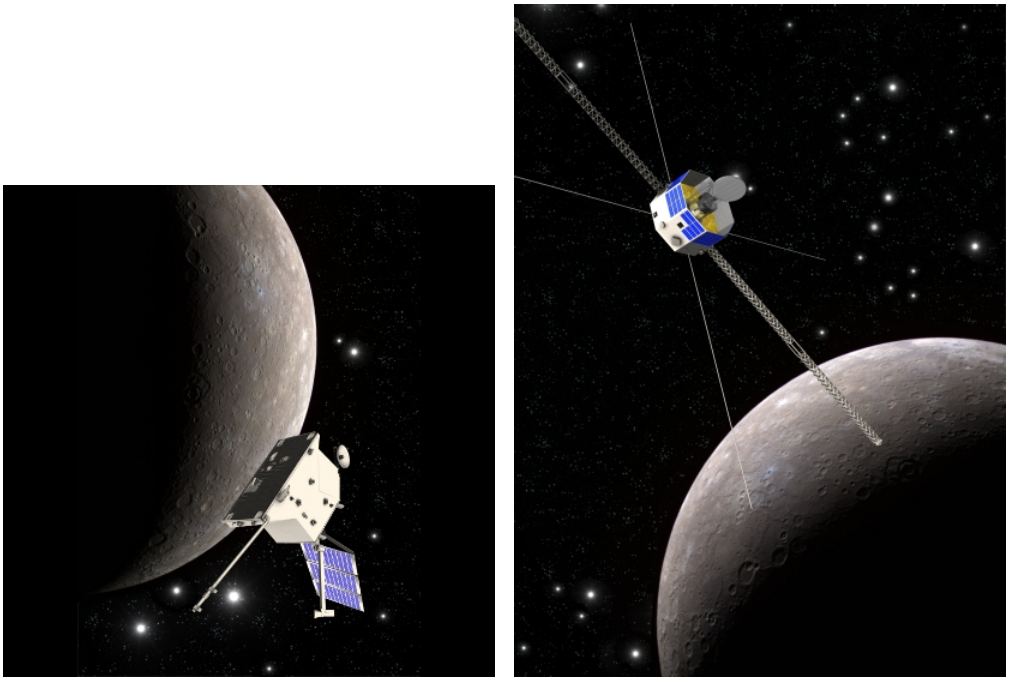
5.1.1 *A short Presentation of BepiColombo*

BepiColombo is mission to the innermost and smallest planet in the Solar System, Mercury, and is jointly conducted by the European Space Agency (ESA) and the Japan Aerospace Exploration Agency (JAXA). The spacecraft, designed and built by EADS Astrium, is currently in the implementation and testing stage. It is due for launch in the summer of 2014 and will take about 6 years to reach its final destination. Mercury is a poorly explored planet, as only two spacecraft to-date have visited the planet. NASA's Mariner 10 and Messenger both flew by Mercury 3 times each, and started in March 2011 a year long science phase in orbit around the planet [NASA, 2011].

The BepiColombo mission consists of two separate orbiters, travelling to Mercury together aboard a composite spacecraft. The Mercury Planetary Orbiter (MPO) will study the surface and internal composition of the planet and the Mercury Magnetospheric Orbiter (MMO) will study Mercury's magnetosphere. The twin orbiters will also investigate the permanently shadowed craters of the polar regions for chemical signatures of sulphur or water ice.

MPO and MMO are stacked-up with the Mercury Transfer Module (MTM), consisting itself of electric propulsion and traditional chemical rocket units. In the current scenario, BepiColombo will use a series of gravity-assist manoeuvres around Earth, Venus and Mercury, in combination with deep space manoeuvres with the thrust provided by SEP. Figure 39 is an illustration of the type of trajectory BepiColombo could soon be flying.

For more information about the BepiColombo mission and the trajectory design, the reader can refer for instance to [ESA, 2011, EADS Astrium, 2011, Jehn et al., 2004, Benkhoff et al., 2010, Yarnoz et al., 2006].



(a) MPO (b) MMO

Figure 38: Artist's impression of the BepiColombo 2 orbiters around Mercury: (a) MPO and (b) MMO (downloaded from ESA's website [ESA, 2011]).

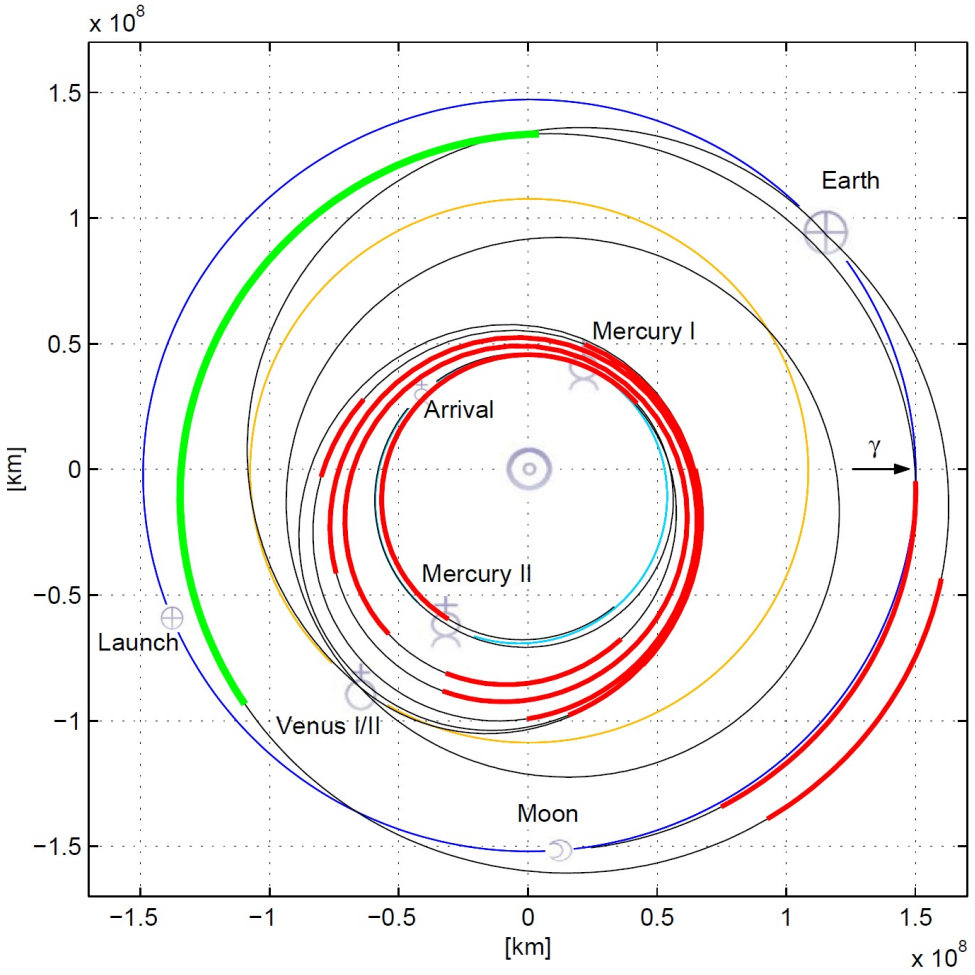


Figure 39: Example trajectory for the BepiColombo mission (from [Jehn et al., 2004]).

5.1.2 *Spacecraft Mass Model*

The mass model presented here is a generic one used for preliminary system mass assessment of a SEP mission. It enables the mass dependence on thrust profile and specific impulse to be evaluated [Kemble, 2006]. The total SEP related mass is given by the following equation:

$$m_{\underset{wet}{SEP}} = m_{tank} + m_{array} + m_{rad} + m_{harness} + m_{PPU} + m_{thrusters} + m_{xenon} \quad (5.1)$$

In this equation, the subsystems considered are the tanks (m_{tank}), the solar arrays (m_{array}), the radiator (m_{rad}), the harness equipment ($m_{harness}$), the power processing unit (m_{PPU}), the thrusters ($m_{thrusters}$) and finally the propellant required to perform the low thrust transfer (m_{xenon}). The expressions of all these quantities are given in the following subsections.

The mass model has been kindly provided by S. Kemble, senior expert in Mission Analysis at Astrium UK, Stevenage. It is based on decades of experience in interplanetary mission design, and is representative to what is used at the preliminary stage of a space mission design. The numerical values of the parameters used in this model have been suggested by Astrium's experts for the relevant subsystem.

Mass of SEP-related Subsystems

TANK The mass of the tank is directly proportional to the mass of propellant:

$$m_{tank} = \sigma_{tank} * m_{xenon} \quad (5.2)$$

where σ_{tank} is the specific ratio of the tank subsystem ($\sigma_{tank} = 11\%$).

SOLAR ARRAYS The area of the solar arrays required is given by equation 5.3.

$$A_{SA} = \frac{P_{1AU}}{\eta_p * G_s} * \kappa_A \tag{5.3}$$

where P_{1AU} is the power to be generated by the solar arrays at 1 Astronomical Unit, η_p is the power conversion efficiency ($\eta_p = 0.22751$), G_s is the solar constant at 1AU ($G_s = 1367$ W.m^{-2}), and κ_A is the area margin for the solar arrays ($\kappa_A = 1.2$).

Using the area of the solar arrays, their mass is given by equation 5.4.

$$m_{array} = \left(A_{SA} * \rho_{SA} + m_{\substack{0 \\ array}} \right) * \kappa_{SA} \tag{5.4}$$

where ρ_{SA} is the specific ratio mass/area of the solar arrays ($\rho_{SA} = 2.89$ kg.m^{-2}), $m_{\substack{0 \\ array}}$ is the inevitable structural mass of the solar arrays (constant and independent of A_{SA}) and κ_{SA} is the mass margin for the solar arrays ($\kappa_{SA} = 1.1$).

RADIATOR The radiator (and the associated elements) is sized based on the maximum power P_{max}, thus at closest approach to the Sun. In the case of the BepiColombo mission, this is at the perihelion of Mercury's orbit, i.e. 0.3 AU. The sizing is performed using a system of two equations that link the power used by the thrusters, the thrust, the specific impulse and the voltage. The power is a linear function of the thrust and the square root of the voltage. The specific impulse on the other hand is a second order polynomial of the trust with linear coefficients of the square root of the voltage. With the

thrust equal to its maximum ($T = T_{max}$) and the specific impulse equal to its value at maximum thrust ($I_{SP} = I_{SP}_{max\ T}$), the voltage is first computed using equation 5.5, then the maximum power P_{max} via equation 5.6.

$$I_{SP} = b_2 T^2 + b_1 T + b_0 \tag{5.5}$$

$$P = \frac{c}{\eta_{engine}} * (a_1 T + a_0) \tag{5.6}$$

where a_1, a_0, b_2, b_1 and b_0 are linear functions of $\sqrt{V}$. V is the voltage in volts, η_{engine} the efficiency of the engine to convert electric power into thrust and c a constant.

Once P_{max} is known, the dissipated power while at perihelion can be evaluated:

$$P_{dis} = \delta_p P_{max} + Q \tag{5.7}$$

where δ_p is the percentage of the maximal power that is wasted ($\delta_p = 0.15$) and Q is the heat to be dissipated at perihelion (constant and independent of P_{max}).

Two different types of radiator can be envisaged for this BepiColombo-like mission. The choice depends on the value of the dissipated power (cf. equation 5.7) being above or below a given threshold P_{dis}_{lim}. The mass of the radiator and its associated structure is calculated using the following equation:

$$m_{rad} = \begin{cases} \left(c_0 + c_1 \frac{P_{dis}}{P_{dis}_{lim}}\right) * \kappa_{rad} & \text{if } P_{dis} < P_{dis}_{lim}, \\ \left(c_2 + c_3 \frac{P_{dis}}{P_{dis}_{lim}} + c_4 \left(\frac{P_{dis}}{P_{dis}_{lim}}\right)^2\right) * \kappa_{rad} & \text{otherwise.} \end{cases} \tag{5.8}$$

where c_0, c_1, c_2, c_3 and c_4 are constants and κ_{rad} is the mass margin for the radiator ($\kappa_{rad} = 1.15$).

HARNESS The harness mass is given by the following equation:

$$m_{harness} = m^0_{harness} + \rho_{harness} P_{max} \kappa_{harness} \tag{5.9}$$

where $m^0_{harness}$ is the inevitable mass of the harness subsystem, $\rho_{harness}$ is the specific ratio mass/power of the harness subsystem ($\rho_{harness} = 1.3763 \cdot 10^{-3}$ kg.W^{-1}) and $\kappa_{harness}$ is the mass margin for the harness subsystem ($\kappa_{harness} = 1.2$).

POWER PROCESSING UNIT The mission of BepiColombo is designed with 4 power processing units (PPU). The mass of each of them is estimated using an equation linear with the maximum power P_{max} (cf. equation 5.6) and the square of the mean specific impulse (cf. equation 5.12).

THRUSTERS Finally, the mass of the thrusters and the associated components varies with the technology used and also the number of thrusters necessary to achieve the required thrust.

$$m_{thrusters} = m^0_{thrusters} + n_{thruster} m_{\substack{nominal \\ thrusters}} \tag{5.10}$$

where $m^0_{thrusters}$ is the inevitable mass of the thrusters subsystem, $m_{\substack{nominal \\ thrusters}}$ is the nominal mass of one thruster and $n_{thruster}$ is the number of thrusters installed aboard the spacecraft ($n_{thruster} = 2$).

Propellant Mass

The mass of xenon is estimated from the ΔV budget using the rocket equation.

$$m_{xenon} = m_{sc}^{0}\left(1 - e^{-\frac{\Delta V}{\overline{I_{SP}} * g_0}}\right) \tag{5.11}$$

where m_{sc}^{0} is the translunar orbit mass, i.e. the wet mass of the spacecraft just after the Earth-Moon system escape (*specific to this mission*, $m_{sc}^{0} = 2400$ kg), g_0 is the gravitational acceleration ($g_0 = 9.80665$ m.s^{-2}), ΔV is the delta V budget for the SEP transfer from the Earth-Moon system escape to the Mercury capture (in m.s^{-1}) and $\overline{I_{SP}}$ is the mean specific impulse of the SEP transfer, given in seconds ($I_{\substack{SP \\ max\ T}}$ is the specific impulse at maximum thrust):

$$\overline{I_{SP}} = 0.989 * I_{\substack{SP \\ max\ T}} \tag{5.12}$$

THE ΔV BUDGET The delta V budget is composed of:

- the deep space ΔV (cf. below),
- the ΔV for second Lunar Gravity Assist: 40 m.s^{-1},
- the ΔV for SAA control: 100 m.s^{-1},
- the ΔV for flyby navigation: 260 m.s^{-1},
- the ΔV for other navigation: 280 m.s^{-1},
- and some contingency: +5% of the deep space ΔV.

The deep space ΔV is a quantity essential to any optimisation of spacecraft design. Indeed, it has a direct impact on the propellant mass (cf. equation 5.11)

and the tank mass (cf. equation 5.2). In the frame of the BepiColombo test case, this value is computationally expensive to obtain and cannot be done fully automatically. Therefore, it is not feasible to consider it within the model as it is. In order to overcome this issue, a surrogate model based on 180 different transfers computed independently for various values of P_{1AU} the power to be generated by the solar arrays at 1 Astronomical Unit (AU) and T_{max} the maximum thrust, has been developed (c.f. Appendix C for a list of the 180 sample points used to build the surrogate model). The use of surrogate models has been widely proposed to speed up the optimisation process, without [Jones, 2001, Hawe and Sykulski, 2007, Sakata et al., 2004, Wang, 2003] or with uncertainties [Eldred et al., 2002, Agarwal et al., 2003, Wang et al., 2005]. A surrogate model has the benefit of significantly reducing the computational time but at the expense of accuracy. Therefore great care should be taken to choose the appropriate model and tune its parameters. Kriging has been selected for this application because it is an exact surrogate (i.e. equal to the ΔV at the observed points), accepting non-uniformly distributed sample points (which allow to densely sample areas of great variation) and finally gives an estimate of the prediction error which could be used to drive the sampling. The DACE package [Lophaven et al., 2002] was used, with a first order polynomial regression model and an exponential correlation model. Figure 40 represents the Kriging model, with the black dots marking the sample points.

A SHORT INTRODUCTION TO KRIGING PREDICTOR The idea of Kriging approximation is to model the lack of knowledge one has in the function value at some non-sampled point $\mathbf{x}$. This is done by considering that the value of the function at $\mathbf{x}$ is "like the realization of a random variable $Y(\mathbf{x})$ that

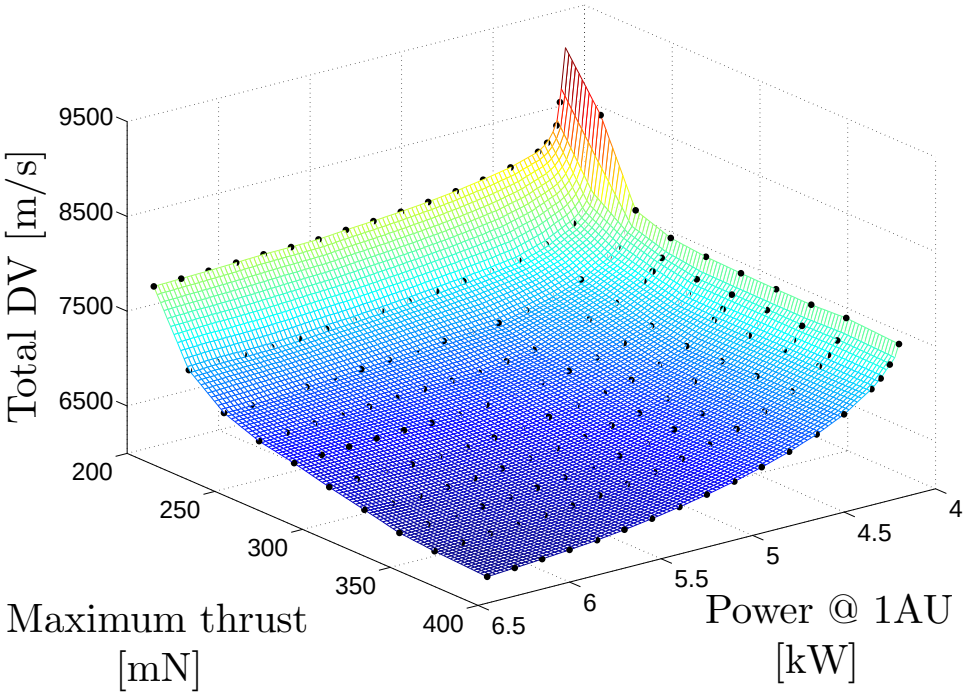


Figure 40: Kriging surrogate of the deep space ΔV for the low thrust mission of BepiColombo.

is normally distributed with mean μ and variance σ^2" [Jones, 2001]. Then, a correlation function is defined by:

$$Corr\Big[Y(x_i), Y(x_j)\Big] = exp\left(-\sum_{l=1}^{N} \theta_l \mid x_{i_l} - x_{j_l} \mid^{p_l} \right) \tag{5.13}$$

where x_i and x_j are two points of the input space of dimensionality N.

The parameters θ_l and p_l, $(l \in [1, N])$ are tuning parameters adjusted such that the model fits the observations. The mean μ and standard deviation σ are also unknown and need to be selected. To do so, the Kriging methodology is to find the tuning parameters that maximise the likelihood of the observed data. Therefore, the definition of the parameters is an optimisation problem itself. To simplify it, some additional properties of the correlation models are chosen. The MatLab toolbox *DACE* [Lophaven et al., 2002] suggested 3 types

of simplified correlation models based on the previous formulation (exponential, general exponential and gaussian) but one can also choose other models such as the one suggested in [Sacks et al., 1989].

For more information about Kriging, the reader can refer to [Cressie, 1990, Davis, 2002]. Also, other alternative meta-models are available such as splines [Bartels et al., 1987] or radial basis functions [Powell, 2001].

Summary

The simple model presented here enables the analyst to estimate the mass of the main subsystems of a low thrust spacecraft with only three inputs: (i) the power to be generated by the solar arrays at 1AU P_{1AU}, (ii) the maximum thrust T_{max} and (iii) the specific impulse at maximum thrust $I_{\underset{max\ T}{SP}}$. An illustration of the variation of the SEP related mass with the design variables is given in Figure 41.

5.1.3 *The OUU problem*

The objective of the design is to maximise the Belief that the SEP-related mass $m_{\underset{wet}{SEP}}$ is as small as possible. The design domain and the BPA-structure are detailed hereafter.

DESIGN VARIABLES The mass modelling is such that only 3 design variables are necessary to define the mass of the SEP-related subsystems. Table 14 gives the range of possible values of each of them.

THE BPA-STRUCTURE In this application, we have selected three parameters as uncertain: the power conversion efficiency η_p, the specific ratio

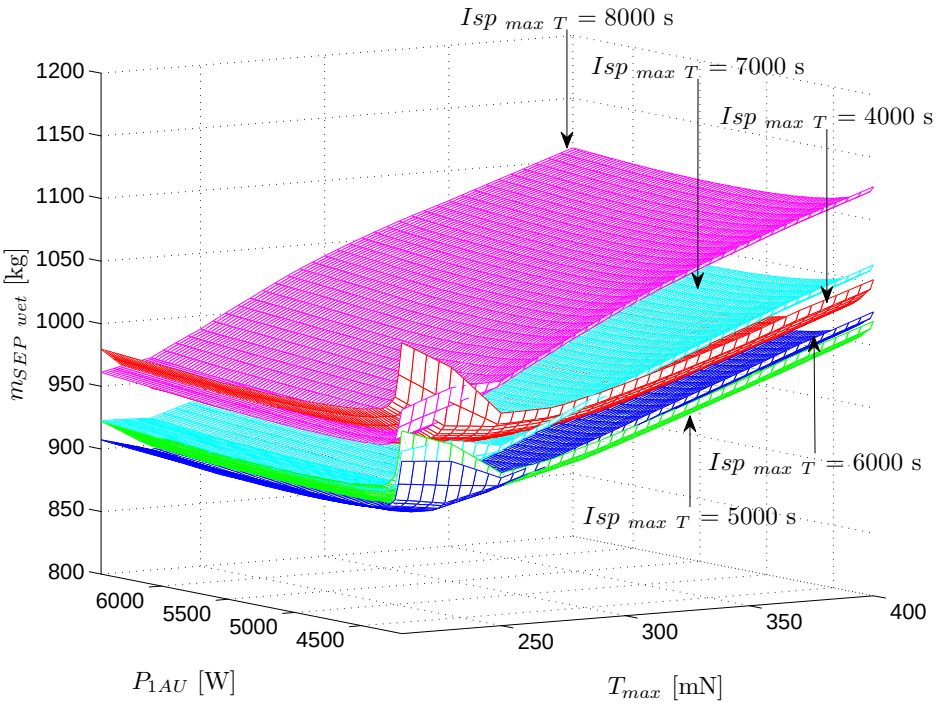


Figure 41: Variation of the SEP wet mass with the design variables for the BepiColombo test case.

Table 14: Design variables for BepiColombo test case.

Parameter	**Symbol**	**Lower bound**	**Upper bound**	**Unit**
Power at 1AU	P_{1AU}	4200	6450	[W]
Maximum thrust	T_{max}	210	400	[mN]
Specific impulse (at maximum thrust)	$I_{SP}{}_{max\ T}$	4000	8000	[s]

mass/area of the solar arrays ρ_{SA}, and the specific ratio mass/power of the harness subsystem $\rho_{harness}$. They appear respectively in equations 5.3, 5.4 and 5.9.

Table 15 presents the BPA-structure. These choices are the result of recommendations from experts at EADS Astrium UK. Indeed, various technologies and quality of space solar power systems are available to the designer, and their performances vary significantly, directly impacting on the value of η_p and ρ_{SA}. Similarly, the specific mass/power ratio of the harness subsystem is dependent on the technology used but also on the internal configuration of the spacecraft, which is unknown at the preliminary stage of the spacecraft design.

Table 15: Uncertainty representation using Evidence Theory - BepiColombo test case.

Uncertain parameter	Intervals		Basic probability assignment
	Lower bound	Upper bound	
η_p	0.18959	0.195	0.05
	0.195	0.205	0.15
	0.205	0.215	0.25
	0.215	0.22751	0.55
ρ_{SA}	2.89	3.00	0.10
	3.00	3.10	0.15
	3.10	3.25	0.35
	3.25	3.3105	0.40
$\rho_{harness}$	$1.3763 \cdot 10^{-3}$	$1.4500 \cdot 10^{-3}$	0.05
	$1.4500 \cdot 10^{-3}$	$1.5500 \cdot 10^{-3}$	0.25
	$1.5500 \cdot 10^{-3}$	$1.6000 \cdot 10^{-3}$	0.30
	$1.6000 \cdot 10^{-3}$	$1.6515 \cdot 10^{-3}$	0.40

A margins approach is typically used to take into account uncertainties when designing space missions. As the goal here is to crystallise the uncertainties and their impact with Evidence Theory, we selected parameters as uncertain when they are associated with a system margin, and set these to 0 in the OUU

problem. In our example, these are κ_A, κ_{SA} and $\kappa_{harness}$. Their values in a conventional design approach are reported in Table 16.

Table 16: Margins applied in the low thrust spacecraft model.

Margins	Value	Subsystem
ΔV	+5%	ΔV contingency
κ_A	1.20	Area of the solar arrays
κ_{SA}	1.10	Mass of the solar arrays
κ_{rad}	1.15	Mass of the radiator
$\kappa_{harness}$	1.20	Mass of the harness subsystem

Note that the BPA-structure is such that the effect of the 3 parameters being considered as uncertain is artificially equivalent to applying the default system margins. The consequence is that the optimal design of the OUU giving a Belief of 1 is the same as the deterministic one. This is obviously not generally the case but it helps here to better comprehend the results.

5.1.4 *Results and Comparisons*

The proposed approaches to solve the OUU problem have been tested against the test case of the BepiColombo mission described previously. It is believed that a (nearly) optimal solution has been identified after very extensive simulations. It serves here as a reference to evaluate the quality of the results found for each test by calculating, as in the previous chapter, the error area between the solution and the optimal curve in the objective space.

The locations of the optimal design points are given in Figure 42. It is important to realise that 2 classes of solutions exist for this problem, distinct by the value of P_{1AU}: 4,650 or 4,800 watts. The optimal maximum thrust is

clearly 230 mN and the specific impulse at maximum thrust between 5639 and 5655 seconds.

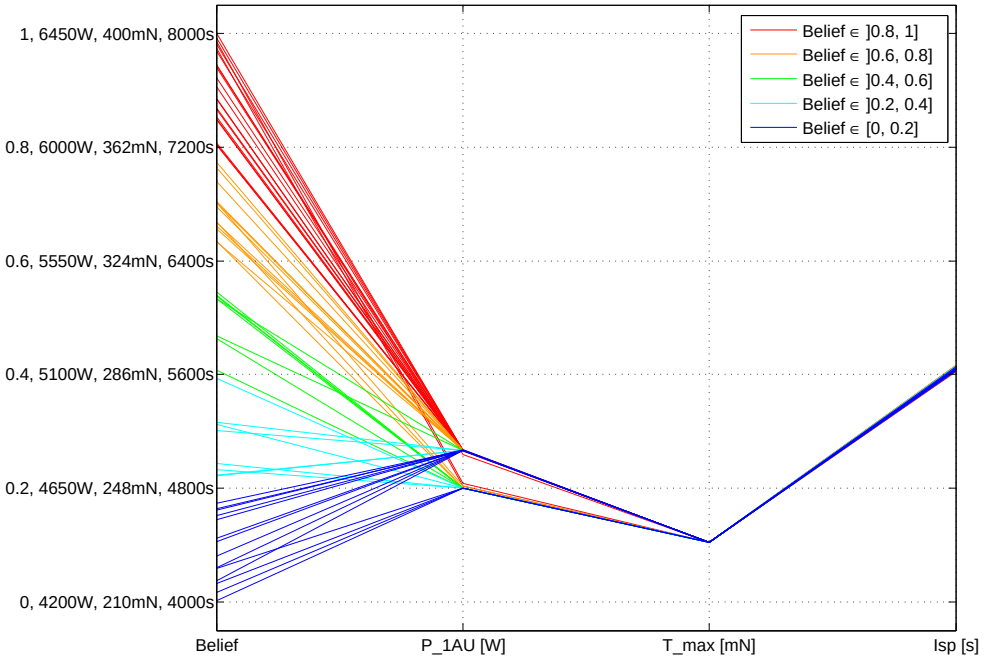


Figure 42: Locations of the optimal design points for different level of Belief - Bepi-Colombo test case.

Direct Solution Simulations

Firstly, the direct solution is tested for three different numbers of total system function evaluations: 100,000 , 500,000 and 1,000,000. Each system function evaluation costs 0.00034 s on an Intel Pentium D, 3.6GHz with 1GB of RAM. As the multi-objective optimiser is not deterministic, 100 simulations have been run for both implementations (bi-objective and multi-belief) to obtain meaningful conclusions. Moreover, the setting of NSGA2 were: (i) agents: 20 (ii) probability of crossover and mutation: 0.75 and 0.33 (iii) distribution index for crossover and for mutation: 10 and 25. As for any test involving an evolutionary algorithm, the settings of the optimiser parameters is tricky and

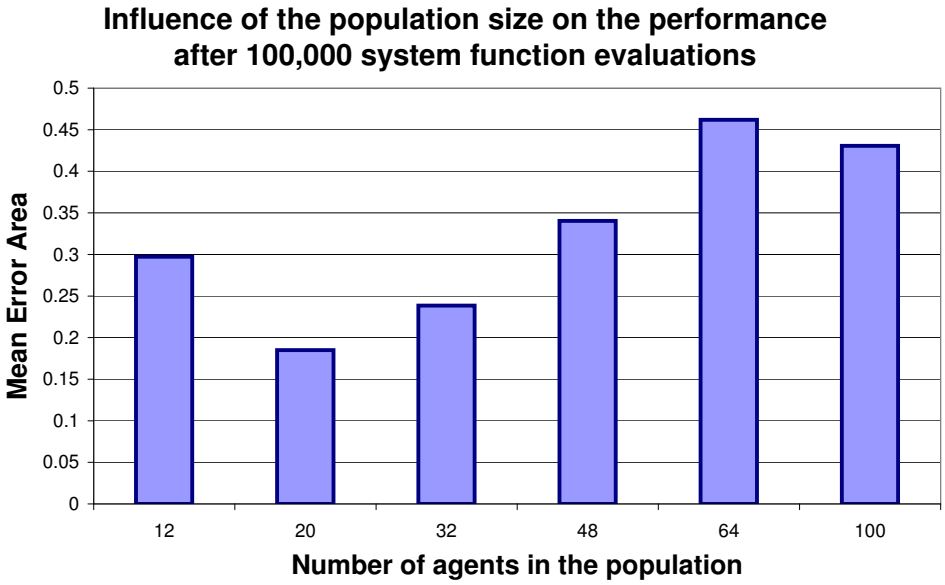


Figure 43: Influence of the number of agents in the performance of NSGA2 for the BepiColombo test case, bi-objective method.

can significantly affect the results. The probabilities and distribution indices were selected such that convergence speed and global exploration is balanced. The most significant parameter however is clearly the size of the population. Preliminary tests for up to 100,000 function evaluations indicated that 20 agents was a suitable population. Figure 43 shows that for the bi-objective function, the optimal population size is around 20 for our selection of probabilities and distribution indices. The same population size was used for the multi-Belief method for comparison purposes.

The BPA structure defined for the BepiColombo-like test case is composed of 64 adjacent focal elements (cf. Table 15). As we do not assume convexity of the system function $m_{\substack{SEP \\ wet}}$, a local optimiser* is used to identify the maximum of the system function over each of the 64 focal elements.

*The Matlab function `fmincon` is used here.

The average solution found for one hundred function evaluations is given in Figure 44, and the mean value of the error area and its variance are given for all simulations in Table 17.

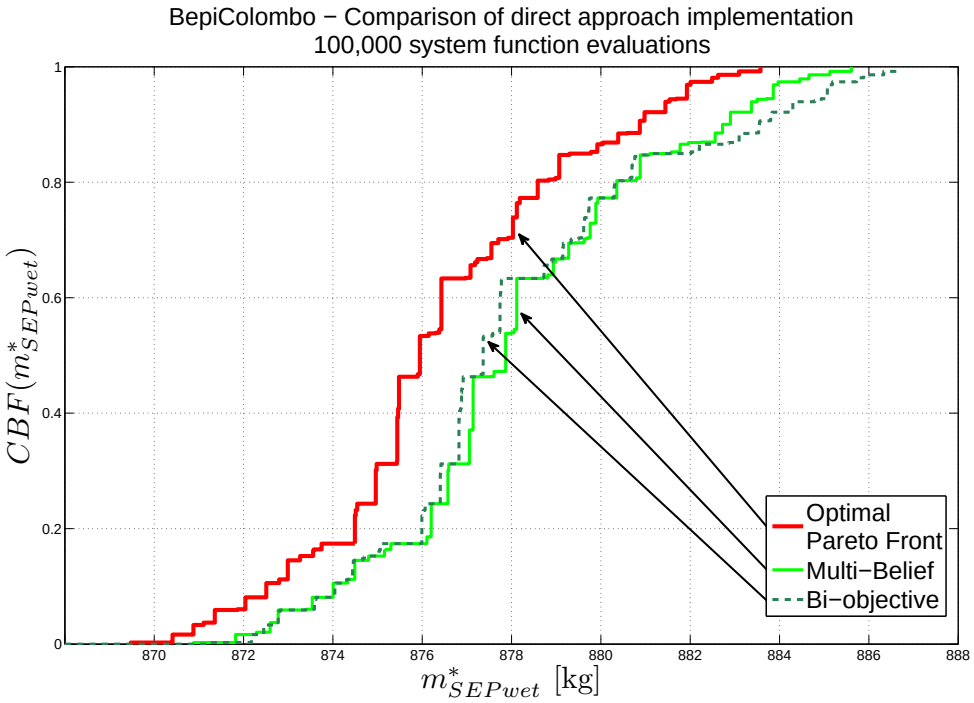


Figure 44: Solutions found for the OUU with only 100,000 system function evaluations - BepiColombo test case.

Table 17: Mean value and variance of the normalised error area for the OUU BepiColombo test case for 100 runs.

n_{val}	Bi-Objective		Multi-Belief	
	mean	variance	mean	variance
100,000	$2.39 \cdot 10^{-1}$	$5.23 \cdot 10^{-2}$	$2.36 \cdot 10^{-1}$	$4.78 \cdot 10^{-2}$
500,000	$9.26 \cdot 10^{-3}$	$2.37 \cdot 10^{-5}$	$9.85 \cdot 10^{-3}$	$1.63 \cdot 10^{-5}$
1,000,000	$5.27 \cdot 10^{-3}$	$2.53 \cdot 10^{-6}$	$3.24 \cdot 10^{-3}$	$3.00 \cdot 10^{-6}$

Finally, the OUU problem can be considered as successfully solved if the classes of optimal design are identified. In the test case of BepiColombo, there are two classes of optimal solutions defined by two boxes:

$$\begin{aligned} \text{Class 1} = &\quad [4640\text{ W}, 4740\text{ W}] \times [229\text{ mN}, 231\text{ mN}] \times [5620\text{ s}, 5680\text{ s}] \\ \text{Class 2} = &\quad [4780\text{ W}, 4820\text{ W}] \times [229\text{ mN}, 231\text{ mN}] \times [5620\text{ s}, 5680\text{ s}] \end{aligned} \tag{5.14}$$

Table 18 gives the rate for which solutions have been found in both classes, or in either class. Both approaches have very similar performances. It is interesting to note the bi-objective approach finds solutions in both classes more often than the multi-belief one. The reason for this is that the bi-objective tests more designs for the same computational effort, therefore increasing, as a consequence, the success rate of finding a solution.

Table 18: Percentage for which solutions have been found over 100 runs in both classes and in at least one class, for the case of BepiColombo.

Number of system	Bi-Objective		Multi-Belief	
function evaluations	both classes	one class	both classes	one class
100,000	2%	20%	0%	2%
500,000	94%	99%	58%	100%
1,000,000	100%	100%	79%	100%

Indirect Solution Simulations

For the indirect approach, 3 different methods have been tested: the clustering - convex hull method, the pixelisation method and the BSP method. The indirect approach is designed to tackle problems with computationally expensive system function. Therefore, to be representative of a real case scenario, the number of system functions has been limited to 100,000. The resulting approximation of the Pareto front is represented in Figure 45. The pixelisation and the BSP methods are such that they may overestimate the real result. This is the reason

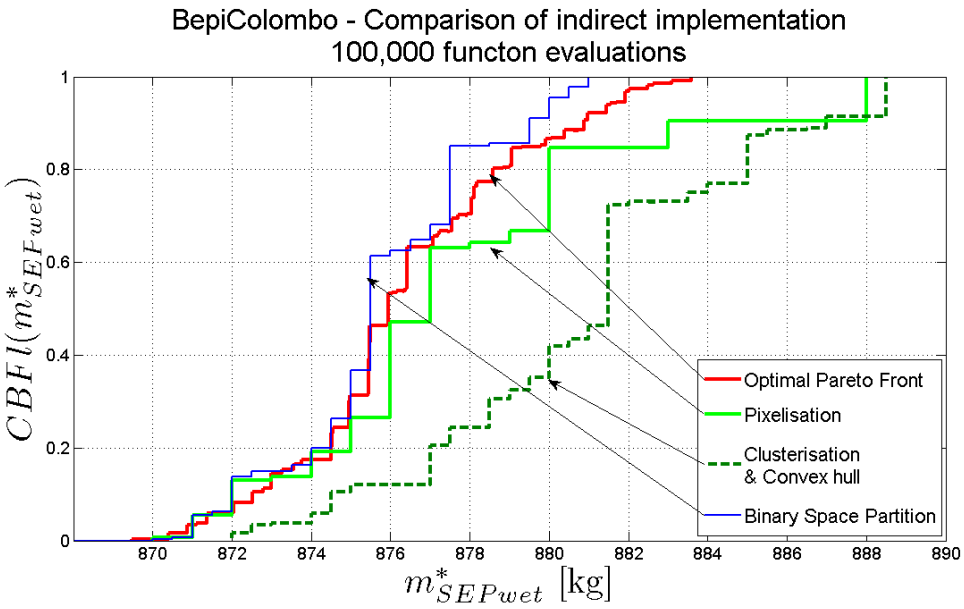


Figure 45: Approximation found with the indirect approaches for the OUU with only 100,000 system function samples - BepiColombo test case.

why they appear to be far better than the clustering - convex hull method in the objective space, and even a little better than the actual solution. The BSP in particular overestimated the actual solution. It remains that all three approximations provide reasonably good approximations of the Pareto front.

In contrast to the direct solution methods, the complexity of the indirect one does not increase exponentially with the number of focal elements. Indeed, only the focal elements that lie between the outer and inner axis-aligned boxes need to be checked. Their number will remain limited and fairly constant throughout the procedure as the inner and outer AABs grow at the same rate. Moreover, the number of sample points needed to gather the same information increases polynomially with the number of dimensions. It is not dependent on the number of focal elements in any way. Figure 46 shows the number of design points that the direct approach can test with 100,000 function evaluations. As the number of focal elements increases, the result of the direct approach

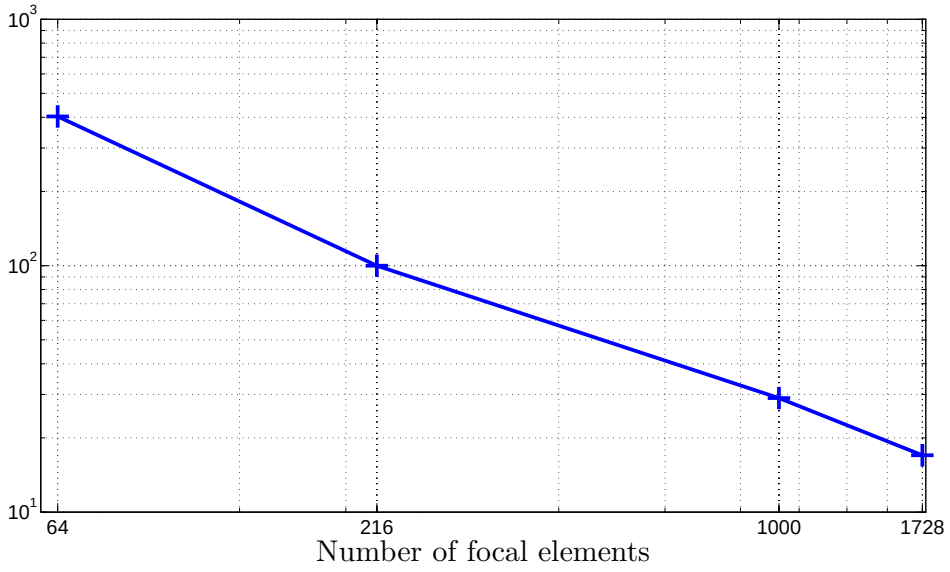


Figure 46: Variation of the number of designs evaluated in the direct approach versus the number of focal elements. The number of system function evaluations has been fixed to 100,000.

naturally decreases in quality. On the other hand this increase will have no effect on the indirect approach.

5.1.5 *Conclusions*

A BepiColombo-like mission has been used as an application to the OUU problem. The direct and indirect methods have been used and compared. The results obtained during this test case confirm the conclusions drawn in the previous chapter when only an analytical test case was used. Note that for this representative test case, it was necessary to substitute with a surrogate model the variation of the ΔV with the design variables. Indeed, the ΔV budget is dependent on the trajectory, and the one BepiColombo will fly is a very complex one indeed. Calculating such a trajectory requires a few minutes of

computational time, and for each trajectory design, the algorithm must be set up properly and verifications must be done on the results to validate them.

5.2 ANOTHER CASE STUDY: MISSION TO C-TYPE ASTEROID

The first space-related test case was strongly inspired from an already well defined mission. For this second test case, the context is more of a feasibility study. The aim is to illustrate the proposed methodology to tackle a reliable preliminary space mission design.

The next section details the proposed test case. First the mission scenario, the trajectory design and the mass modelling are described. The OUU problem is then formulated.

The Binary Space Partition method is used as a first step to gather basic knowledge of the problem at hand. The bi-objective method is then used on the pruned search space to refine the results.

5.2.1 *Presentation*

Mission scenario

In this test case, the objective is to design a reliable mission to rendez-vous with the asteroid 1999JU3. The departure from Earth is scheduled for 2020. The 10 months period from the 1st of January 2020 (7304.5 MJD2000) until the 1st of November 2020 (7609.5 MJD2000) is favourable for a transfer.

The time of flight is also constrained from 280 to 650 days. Time of flight lower than 280 days would require impractical levels of ΔVs, while an upper

bound of about two years is representative of constraints linked to biological payloads or operational costs for instance.

Finally, the wet mass at launch is fixed to $m_{sc}^0 = 600$ kg. Therefore, the objective of the design is in this case to maximise the mass of payload m_{PL}.

Asteroid 1999JU3

The asteroid 1999JU3, also designated as 162173 [Campins et al., 2009], is the target for a second sample return mission to primitive bodies by the Japanese space agency JAXA, after the success of Hayabusa launched in 2003. Hayabusa was a sample return mission to the S-type asteroid Itokawa, pictured in Figure 47 in 2005. Hayabusa-2 has been confirmed for a launch in 2014 or 2015, and should return collected samples from 1999JU3 in the year 2020 [JAXA, 2011].

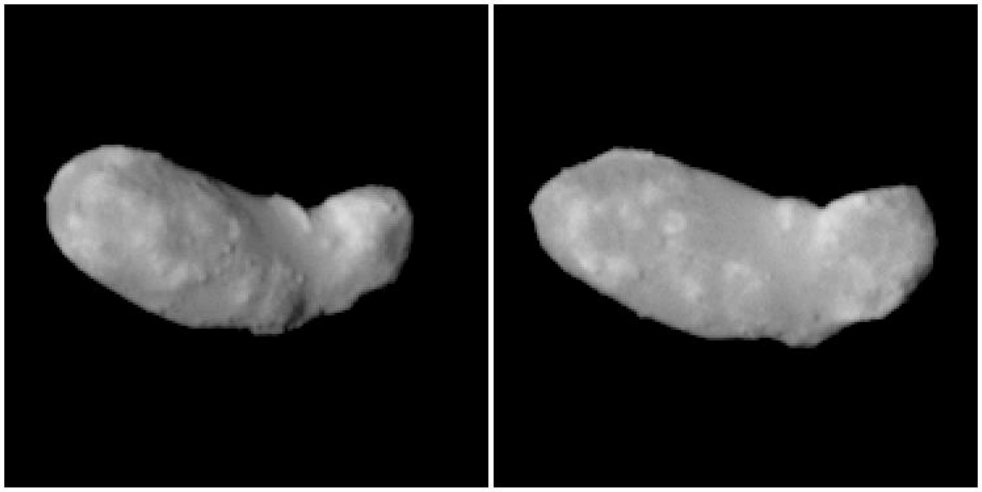


Figure 47: Pictures taken by Hayabusa of the asteroid Itokawa, on the 10/09/2005 [Spacecraft, 2005].

The asteroid 1999JU3 is classified as a C-type asteroid, which are known to be parent bodies of carbonaceous chondrites, having unaltered composition and more organic matters than other meteorites [Hasegawa et al., 2008]. C-type asteroids are the most commonly known ones (∼75%), have generally a low

albedo (less than 0.04, thus very dark), and are distributed mostly around the outer side of the asteroid belt. The other main types of asteroids according to Tholen's taxonomy [Tholen, 1989] are S-type (∼17% of known asteroids, moderately bright (albedo ∼0.14) and mainly metallic nickel-iron mixed with iron- and magnesium-silicates) and the M-type (most of the rest, moderately bright (albedo of 0.10 to 0.18), with a surface composition of metal, olivine and pyroxene).

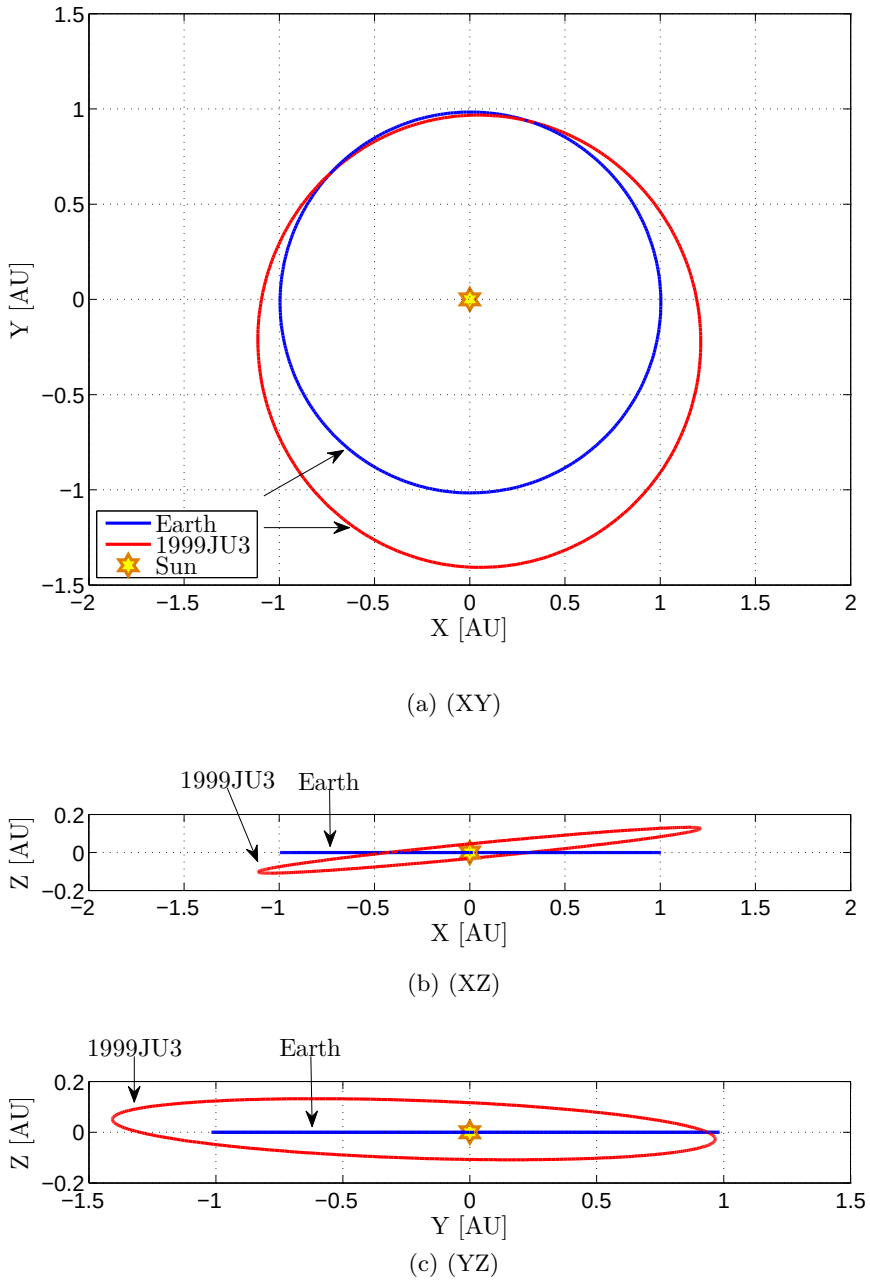
With an apocenter radius of 1.4AU and a pericenter radius of 0.96AU, 1999JU3 is classified as an Apollo type asteroid, as it crosses the Earth's orbit. Also, the synodic period of the Sun-Earth-1999JU3 system is 4.37 years. The keplerian orbital elements of the asteroid are given in Table 19 and its orbit is represented in the ecliptic plane in Figure 48.

Semi major axis [AU]	**Eccentricity**	**Inclination** [deg]	**Right Ascension of the Ascending Node** [deg]	**Argument of periapsis** [deg]
1.1891	0.18996	5.8842	251.71	211.29

Table 19: Keplerian orbital elements of the asteroid 1999JU3 used in this study (updated keplerian elements are made available regularly at the NEODYS of the University of Pisa `http://newton.dm.unipi.it/neodys`).

Trajectory Design Through Shape-Based Approach

As is the case for almost all space exploration missions, the trajectory is a key element of the design process. This is even more true when a low thrust propulsion system is used as it has an impact not only on the propellant and tank mass but also on the power requirements which in turn affect the size of the solar arrays, the radiator, harness and power processing units. Moreover,



(a) (XY)

(b) (XZ)

(c) (YZ)

Figure 48: Orbit of the Asteroid 1999JU3 in (a) the ecliptic plane and (b)-(c) out-of-plane.

low thrust engines can produce by definition only a very low amount of thrust, typically values of a few mN, that the trajectory should be compliant with.

When the mission design is at a preliminary stage, a computationally efficient way to calculate a trajectory is required to enable the investigation of as many alternatives as possible.

In this test case, a shape-based method to compute the trajectories and the associated controls has been used. The implementation of this method presented in [Novak and Vasile, 2009] has been kindly made available for this work by the lead author Dr D. Novak.

To summarise, the shape-based approach consists in expressing the spacecraft's position with the pseudo-equinoctial elements and using the longitudinal anomaly L for parametrisation [Novak and Vasile, 2009]. The expression of the pseudo-equinoctial elements are given in equation 5.15. For more information, please refer to [Novak and Vasile, 2009].

$$\begin{cases} p = p_0 + p_1 e^{\lambda_1 (L - L_0)} \\ f = f_0 + f_1 e^{\lambda_2 (L - L_0)} \\ g = g_0 + g_1 e^{\lambda_2 (L - L_0)} \\ h = h_0 + h_1 e^{\lambda_3 (L - L_0)} \\ k = k_0 + k_1 e^{\lambda_3 (L - L_0)} \end{cases} \tag{5.15}$$

When using the pseudo-equinoctial elements to shape the trajectories, the computational time was reduced by setting λ_2 and λ_3 to 0.1 as in [Novak and Vasile, 2009]. Indeed, since the departure and arrival orbits are nearly circular (eccentricity of 0.01675 and 0.18996 respectively), the shaping parameter λ_2 does not substantially affect the time of flight and the total ΔV. With the inclination

of the target 1999JU3 being very low (5.88 deg), the optimal transfers will be close to planar, and therefore, fixing λ_3, the shaping parameters governing the out-of-plane motion, will not affect the results.

The transfer time t should also be given as a function of L but it is difficult to provide a priori such an expression which results in practically acceptable thrust profiles [Novak and Vasile, 2009]. Here, it is shaped by using its derivative with respect to L for the unperturbed orbit, neglecting the control term (cf. equation 5.16).

$$t(L) = \int_{L_0}^{L} \frac{1}{\sqrt{\mu p(l)}} \left(\frac{p(l)}{1 + f(l)\cos L + g(l)\sin L} \right)^2 \mathrm{d}l \tag{5.16}$$

where μ is the gravitational constant of the central body (of the Sun in our case). Of course, shaping the time in such a way leads to a unique time profile directly linked to the shape of the pseudo-equinoctial elements, and therefore to a unique time of flight. As the trajectory has to leave from Earth and arrive at 1999JU3 with no relative velocity, a time of flight constraint has to be satisfied. The shaping parameter λ_1 is thus chosen accordingly via a Newton loop. If it occurs that the time of flight constraint cannot be fulfilled that way, a two step procedure is used:

1. λ_1 is chosen such that the time of flight violation ToF_{viol} is as close as possible to zero: The last two values of λ_1 returned by the Newton loop are on both side of the minimum. They are used as starting points for a bisection on the derivative of the time of flight violation with respect to λ_1, i.e. violation $\frac{\mathrm{d}ToF_{viol}}{\mathrm{d}\lambda_1}$. This leads to λ_1^* for which $\frac{\mathrm{d}ToF_{viol}}{\mathrm{d}\lambda_1}(\lambda_1^*) = 0$.

2. Once λ_1^* is found, the time evolution is shaped as proposed in [Novak and Vasile, 2009], section 1.4. Using the time of flight violation ToF_{viol}^* associated with λ_1^*, the time as a function of L becomes:

$$t(L) = \int_{L_0}^{L} \frac{1}{\sqrt{\mu p(l)}} \left(\frac{p(l)}{1 + f(l)\cos L + g(l)\sin L} \right)^2 + ToF_{viol}^* \frac{6(L - L_0)(L - L_f)}{L_f - L_0} \mathrm{d}l \quad (5.17)$$

Note that the proposed way to deal with the time of flight constraint is a slight modification from [Novak and Vasile, 2009]. The reason is to have results that are as continuous as possible by removing numerical discontinuities. This is critical as the trajectory computation will be part of the objective function in the optimisation under uncertainty problem.

Thrust Profile and Propellant Consumption

Once a trajectory and its associated control $\mathbf{c}(t)$ are known, the second step is to compute the variation of mass along the trajectory.

The thrust magnitude at a given instant is simply the product of the mass of the spacecraft m_{sc} and the norm of the control:

$$T(t) = m_{sc}(t) * \|\mathbf{c}(t)\| \quad (5.18)$$

By replacing T in the relation between the specific impulse and the thrust (c.f. §5.1.2) with the previous equation, we have:

$$I_{SP} = b_2 \left(m_{sc} * \|\mathbf{c}\|\right)^2 + b_1 \left(m_{sc} * \|\mathbf{c}\|\right) + b_0 \quad (5.19)$$

Additionally, let us remember the definition of the specific impulse:

$$\begin{aligned} I_{SP} &= -\frac{T}{g_0 \dot{m}_{sc}} \\ &= -\frac{m_{sc} * \|\mathbf{c}\|}{g_0 \dot{m}_{sc}} \end{aligned} \tag{5.20}$$

where g_0 is the gravitational acceleration at sea level and $\dot{m}_{sc}$ is the mass flow rate.

Combining the two equations 5.19 and 5.20, we find:

$$\begin{aligned} \frac{m_{sc} * \|\mathbf{c}\|}{g_0 \dot{m}_{sc}} &= b_2 \left(m_{sc} * \|\mathbf{c}\|\right)^2 + b_1 \left(m_{sc} * \|\mathbf{c}\|\right) + b_0 \\ \dot{m}_{sc} &= \frac{m_{sc} * \|\mathbf{c}\|}{g_0 (b_2 \left(m_{sc} * \|\mathbf{c}\|\right)^2 + b_1 \left(m_{sc} * \|\mathbf{c}\|\right) + b_0)} \end{aligned} \tag{5.21}$$

As the initial mass m_{sc}^0 is known, we can numerically integrate the differential equation equation 5.21 using, for example, the function `ODE45` of MatLab. This leads to the variation with time of the mass $m_{sc}(t)$ of the spacecraft along the trajectory. Subsequently, the thrust profile can then be reconstructed via equation 5.18.

Note that the design is limited to only one engine with a maximum thrust $T_{max} = 250$ mN. If the trajectory requires a higher thrust level, the trajectory is discarded.

Mass Modelling

PAYLOAD MASS As mentioned in the mission scenario §5.2.1, the objective of the design is to maximise the payload mass. It simply corresponds to the mass of the spacecraft minus the mass of all the equipment needed to fly it to the targeted destination.

$$m_{PL} = m_{sc}^{0} - (m_{struct} + m_{tank} + m_{thrusters} + m_{xenon} + m_{PPU} + m_{array} + m_{rad} + m_{harness}) \quad (5.22)$$

The mass modelling of the subsystem used in this test case is almost identical to the one used for the BepiColombo test case (c.f. §5.1.2). Only 1 thruster and 1 PPU are used ($n_{thruster} = 1$, $n_{PPU} = 1$) and the initial wet mass is much smaller (600 instead of 2400 kg). Also, as the payload mass is optimised, the mass of the structure m_{struct} needs to be taken into account. We fixed it to be 5% of the wet mass at launch, i.e.:

$$m_{struct} = 0.05 * m_{sc}^{0} \quad (5.23)$$

COMPUTING THE POWER AND SPECIFIC IMPULSE In order to use the mass model, the maximum power P_{max}, the power to be generated by the solar arrays at 1AU P_{1AU} and the mean specific impulse $\overline{I_{SP}}$ needs to be evaluated.

The power profile along the trajectory is computed from the thrust profile (c.f. §5.2.1), and using the relation between the thrust and the power required given in equation 5.6. Then, the maximum power P_{max} is available, necessary to calculate the mass of the radiators, harness and PPU subsystems. Also available is the power at 1AU P_{1AU} sizing the solar arrays.

In the BepiColombo test case, the mean specific impulse was computed based on an empirical formula (c.f. equation 5.12). This was because the trajectory and therefore the control law associated was not computed for each design

point. In the present test case however, the control law is available. Therefore, the rocket equation is used as follows to compute the mean specific impulse:

$$\overline{I_{SP}} = \frac{\Delta V}{g_0 * ln\left(\frac{m_{sc}^0}{m_f}\right)} \tag{5.24}$$

The total ΔV of a given trajectory is the integral of the control over the trajectory:

$$\Delta V = \int_{t_0}^{t_0+ToF} \|\mathbf{c}(\tau)\| \, \mathrm{d}\tau \tag{5.25}$$

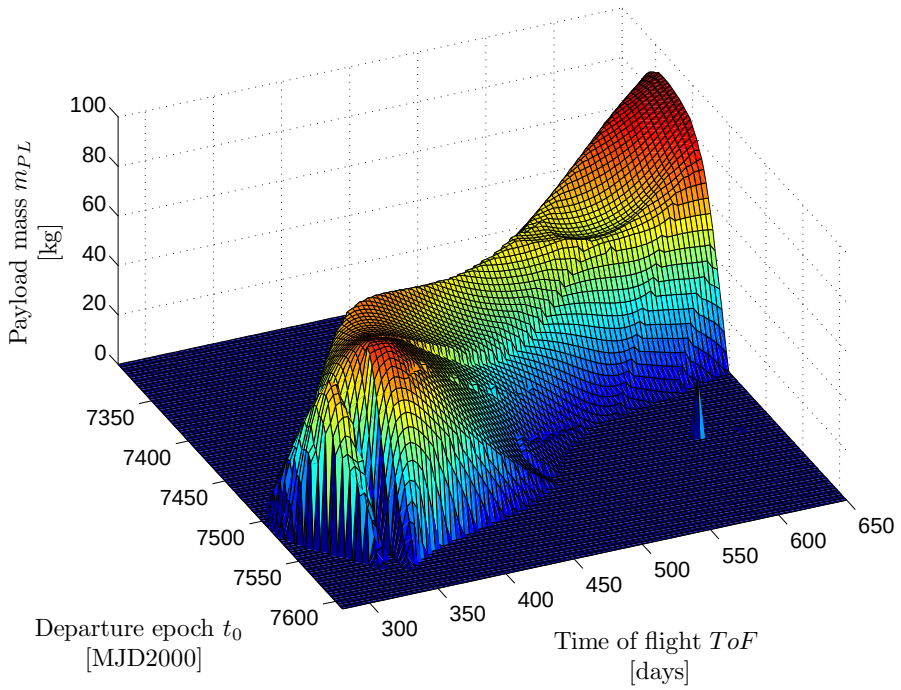
(t_0 is the initial epoch of the trajectory and ToF the time of flight).

Therefore, once the trajectory has been computed, the required information to compute the mass of the different subsystems, and finally of the payload, is available. Figure 49 represents the variation of the payload mass with the departure epoch t_0 and the time of flight ToF (the uncertain parameters have been fixed at their nominal values). The figure clearly show a horse-saddle shape, with two local optima found for an early departure and long time of flight, or a late departure and a short time of flight.

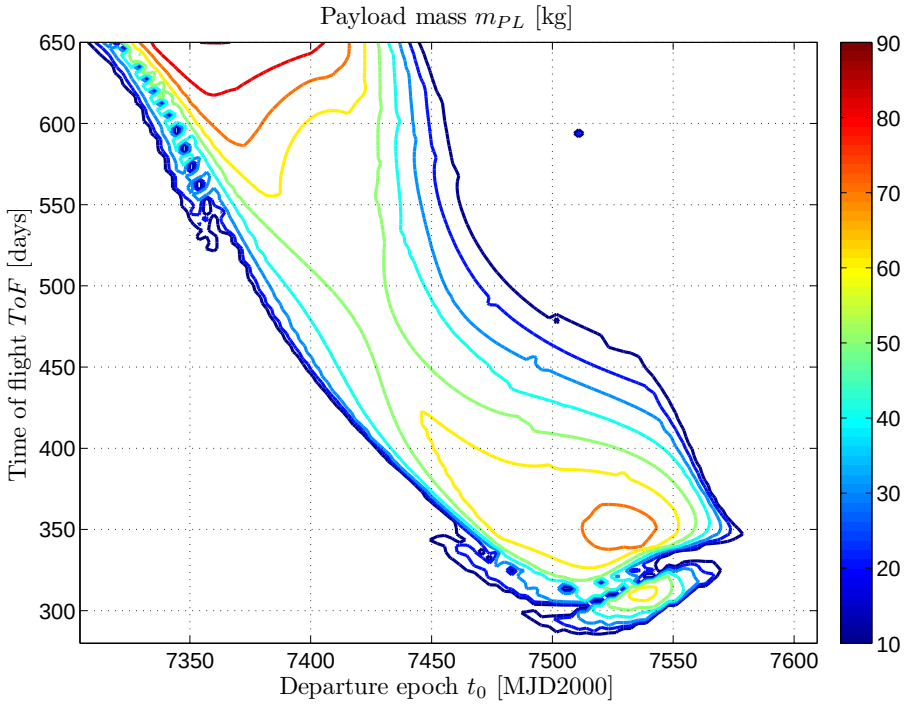
5.2.2 *The OUU Problem*

In contrast to the BepiColombo test case, we aim here at maximising the Belief that the payload mass is greater than a given threshold. Therefore, the Complementary Cumulative Belief Function (CCBF) must be used instead of the CBF.

DESIGN VARIABLES As seen in the previous section, the payload mass is fully defined once the trajectory is available. Therefore, there are in this case



(a) 3D plot



(b) Contour plot

Figure 49: Payload mass to the Asteroid 1999JU3 versus the departure epoch and the time of flight for the deterministic case.

only two design variables, the departure time t_0 and the time of flight ToF. Table 20 gives the range of possible values of the design variables.

Table 20: Design variables for the asteroid mission test case.

Parameter	**Symbol**	**Lower bound**	**Upper bound**	**Unit**
Departure epoch	t_0	7304.5	7609.5	days (MJD2000)
Time of flight	ToF	280	650	days

UNCERTAIN PARAMETERS Five parameters are considered as uncertain. This is two more than the BepiColombo test case, and therefore the problem is expected to be more difficult to solve. The uncertain parameters are:

- η_{engine} the efficiency of the engine to convert electric power into thrust (c.f. equation 5.6)
- η_p the power conversion efficiency (c.f. equation 5.3)
- ρ_{SA} the specific ratio mass/area of the solar arrays (c.f. equation 5.4)
- $\rho_{harness}$ the specific ratio mass/power of the harness subsystem (c.f. equation 5.9)
- δ_p the percentage of the maximal power that is wasted (c.f. equation 5.7)

The BPA-structure, given in Table 21, has a total of 1024 focal elements.

REMARKS This test case is supposed to represent a real life situation in the industry, where an engineer is investigating the feasibility of a space mission. Therefore, nothing is assumed to be known of the optimal solution. Also, a solution should be available within a few hours on a mid-of-the-range

Table 21: Uncertainty representation using Evidence Theory - Asteroid test case.

Uncertain parameter	Intervals		Basic probability assignment
	Lower bound	Upper bound	
η_{engine}	0.735	0.745	0.20
	0.745	0.755	0.40
	0.755	0.760	0.30
	0.760	0.765	0.10
η_p	0.18959	0.195	0.05
	0.195	0.205	0.15
	0.205	0.215	0.25
	0.215	0.22751	0.55
ρ_{SA}	2.89	3.00	0.10
	3.00	3.10	0.15
	3.10	3.25	0.35
	3.25	3.3105	0.40
$\rho_{harness}$	$1.3763 \cdot 10^{-3}$	$1.4500 \cdot 10^{-3}$	0.05
	$1.4500 \cdot 10^{-3}$	$1.5500 \cdot 10^{-3}$	0.25
	$1.5500 \cdot 10^{-3}$	$1.6000 \cdot 10^{-3}$	0.30
	$1.6000 \cdot 10^{-3}$	$1.6515 \cdot 10^{-3}$	0.40
δ_p	0.10	0.12	0.10
	0.12	0.14	0.20
	0.14	0.16	0.40
	0.16	0.18	0.30

personal computer, in order to allow multiple iterations for a trade-off analysis for example.

The OUU problem is more challenging than the BepiColombo case as the number of focal elements is 16 times larger, and there are a total of 5 uncertain parameters instead of only 3. Moreover, the trajectory is computed for each new set of design variables. Even though the shape-based method is very fast, the CPU time required is about one second. Note that the uncertain parameters

do not influence the trajectory. Therefore, when the direct method is used, the trajectory is computed only once per design point, before the mass modelling is applied during the Belief evaluation. The BPA-structure and the system function are such that the extremum lies on one of the 32 vertices of each focal element. Therefore, there is no need to use a local optimiser, and the system function is only evaluated at the vertices to extract the minimal value of the payload mass. The number of function evaluations, to calculate the complete CCBF curve of a single design, however, remains large, 3125 precisely.

Based on these considerations, the methodology used here to solve the OUU problem is: first, the BSP method is used to extract the major trends of the problem, the solution is then refined using the bi-objective method. The methodology follows the conclusions of the previous chapter 4. The bi-objective method is used rather than multi-objective one because the number of focal elements is important. The results obtained at both steps are given in the next sections.

5.2.3 *Using the BSP Method to Identify Promising Design Regions*

The BSP is used first to identify design regions of interest. The pool sampling method has been preferred to the LHSU. Indeed, it is generally the case that a significant share of the couples (departure epoch - time of flight) are not compatible with a reasonable ΔV. Thus, a lot of design points are likely to be infeasible. Pool sampling should help to limit the sampling in the infeasible region.

The setting used for the pool sampling is the same as in the Chebyquad test cases, and given in Table 22. The pool threshold was set to 5% of the departure mass, i.e. 30 kg, which is purposely a bit lower than the desired ratio

(usually 10-20%). Finally, 100 generations of 100 agents are used, thus a set of 10,000 sample points is created. A large population size was chosen here to favour exploration of the design and uncertain parameter space. The number of generations was driven by the time constraint imposed. Indeed, the system function takes a little less than one second to evaluate, thus a little more than 2 hours and a half is required for 10,000 sample points.

Table 22: Setting parameters of NSGA2 for the pool sampling for the mission design to 1999JU3.

Parameter	**Value**
Pool threshold	30
Population size	100
Number of generations	100
Probability of crossover	0.9
Probability of mutation	1/7
Distribution index of crossover	20
Distribution index of mutation	40

Once the sample points are available, the approximated Belief curve is computed. The initial threshold is selected to the highest payload mass found during the sampling, The threshold step is selected to be 1 kg, considered small enough for the accuracy required at a preliminary design stage. Figure 50 gives the approximated optimal Belief curve found.

Figure 51a shows the location of the best design regions for small payload mass and high Belief, and Figure 51b the best design regions for large payload mass and low Belief.

It clearly appears that 3 distinct regions are of interest for the higher values of the payload mass, while there is only 1 for the lower values. This latter is in fact included in one of the design regions identified for higher payload mass. Table 23 gives the best design regions found.

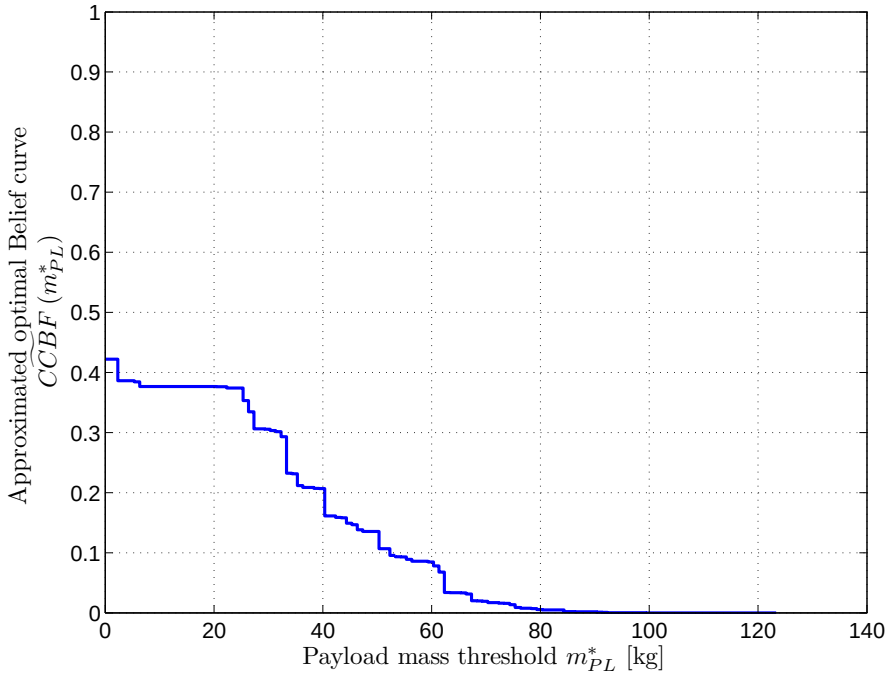
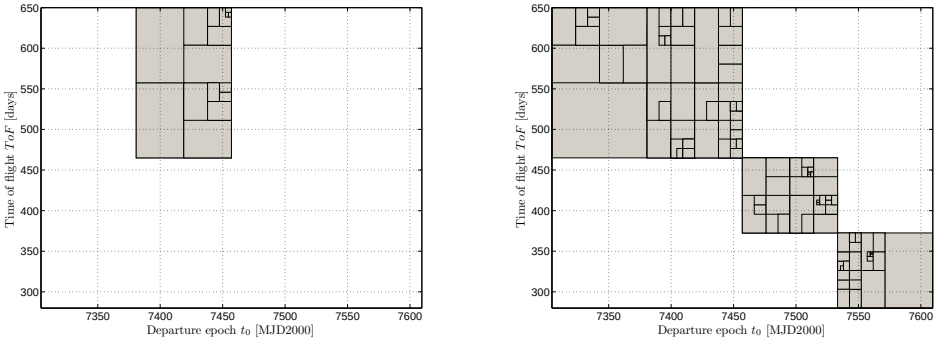


Figure 50: Approximated optimal Belief curve of the OUU problem - Mission to 1999JU3 test case.



(a) $m^*_{PL} \leq 40$ kg (b) $m^*_{PL} > 40$ kg

Figure 51: Best design regions found by the BSP method on the mission to 1999JU3 test case: (a) for low payload mass and (b) for large payload mass.

Table 23: Best design regions identified with the BSP method - Mission to 1999JU3 test case.

Best Design Regions	**Departure epoch** [MJD2000]		**Time of flight** [days]		**Found for**
	Lower bound	**Upper bound**	**Lower bound**	**Upper bound**	
BDR #1	7380.75	7457.00	465.0	650.0	$m^*_{PL} \leq 40$ kg
BDR #2	7304.50	7457.00	465.0	650.0	$m^*_{PL} > 40$ kg
BDR #3	7457.00	7533.25	372.5	465.0	$m^*_{PL} > 40$ kg
BDR #4	7533.25	7609.50	280.0	372.5	$m^*_{PL} > 40$ kg

The CCBF has been computed for the design points at the centre of the 2$^{\text{nd}}$, 3$^{\text{rd}}$ and 4$^{\text{th}}$ design regions. Figure 52 shows these 3 curves. Note that there is a significant difference with the approximated Pareto front obtained with the BSP (c.f. Figure 50). More sample points would have certainly improved the approximation accuracy, and reduced the size of the design regions as well.

Of the 3 design regions being tested, BDR #4 clearly appears the least promising, as there is not even complete certainty that any payload could be brought to destination. BDR #2 and BDR #3 on the other hand are designs that can bring respectively 30 and 50 kg of payload with certainty, and close to 100 kg if decision makers are willing to take significant risks.

5.2.4 *Using the Bi-objective Method to Refine the Best Designs*

The bi-objective method is now used to refine the preliminary results obtained with the indirect method. Three distinct design regions have been identified as interesting, BDR #2, BDR #3 and BDR #4. Thus, the bi-objective method is run 3 times, each time limiting the search space to one of the design regions. The initial Pareto front is set to the CCBF curve computed at the centre of the

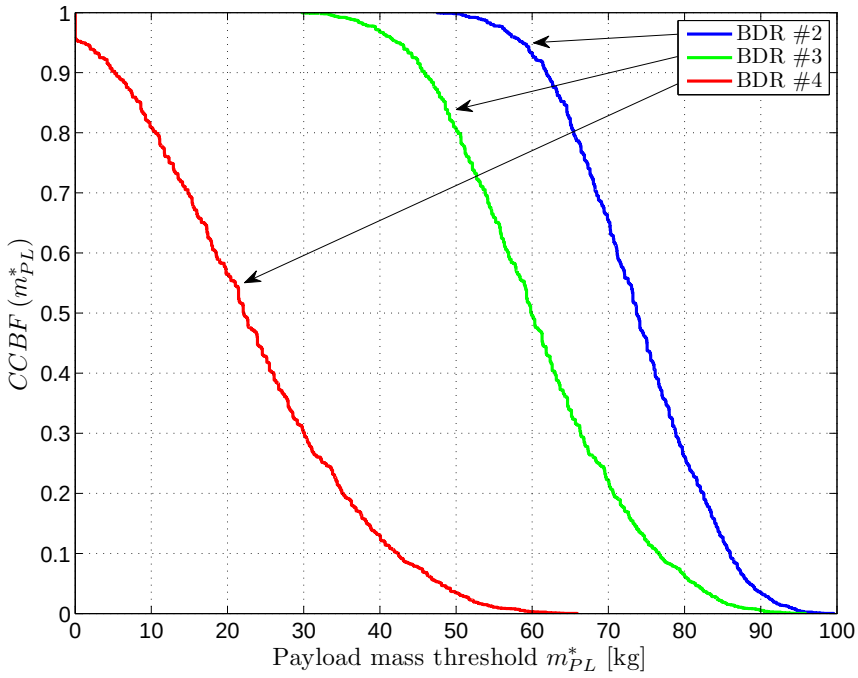


Figure 52: CCBF curves for the centre of the 3 design regions BDR #2, BDR #3 and BDR #4 - Mission to 1999JU3 test case.

corresponding design region (c.f. Figure 52). The evolution of the area between the axis $m^*_{PL} = 0$ and the current Pareto front is plotted for each design region in Figure 53. After 1 million function calls, the bi-objective method does not

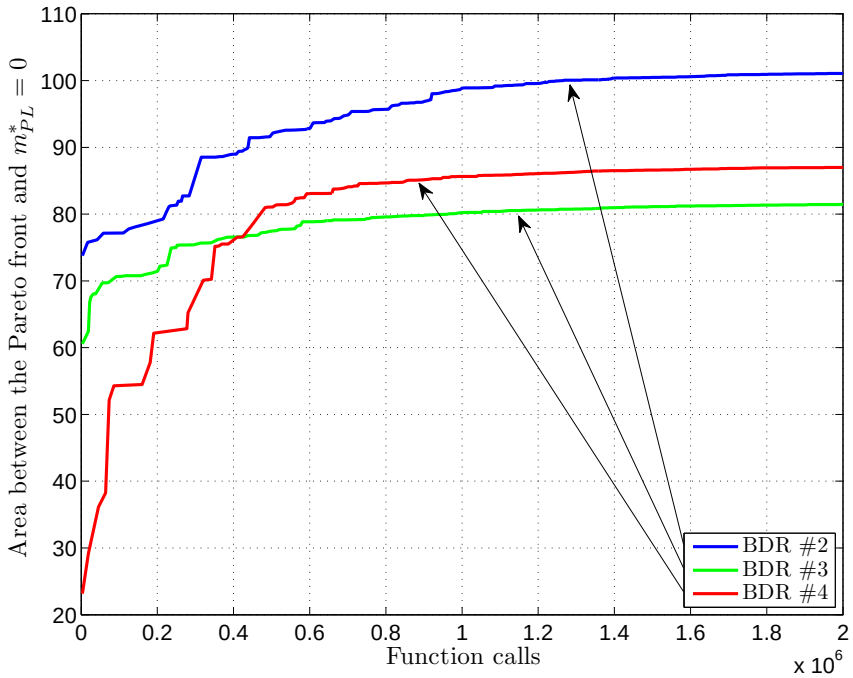


Figure 53: Area versus number of function calls during the bi-objective optimisation for the Mission to 1999JU3 test case (area between the axis $m^*_{PL} = 0$ and the current Pareto front).

significantly improve the results. The code has been left running for another 1 million function calls, at which point convergence was considered reached.

It is interesting to note that even though the design region BDR #4 was the least promising of the three, it turned out that a better result than for BDR #3 was found once refined with the bi-objective method.

5.2.5 *Results and Comments*

The bi-objective method was used to refine the results obtained by the BSP, confining the search space to 3 independent design regions. Each of them converged to different local optimal designs, which are dominating for any belief level. This can be seen in Figure 54. Note that for BDR #3 and #4, the

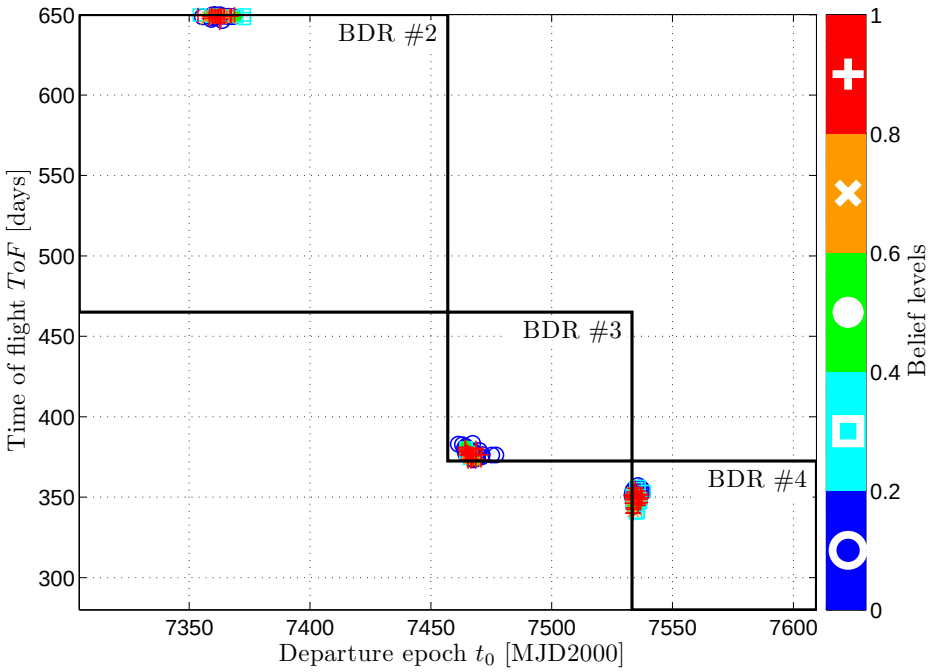


Figure 54: Locations of the optimal design points for the OUU - Mission to 1999JU3 test case.

optimal design is found at the edge of the search space. This tends to indicate that the optimum might be outside, and therefore not reachable. To refine even further the results, one could run the bi-objective method on the design region $\{t_0 \in [7450, 7550];\ ToF \in [325, 400]\}$.

Finally, the optimal Belief curves for the 3 design regions are given in Figure 55. The design found in BDR #2 significantly dominates the two others.

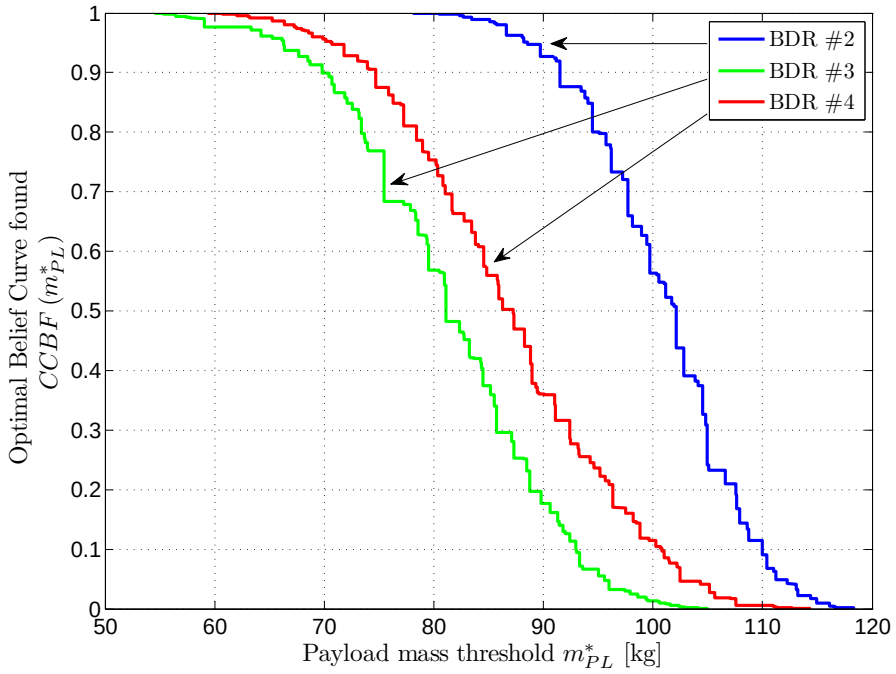


Figure 55: Optimal Belief found in each design region by the bi-objective method - Mission to 1999JU3 test case.

This is much more apparent for high Belief levels (about 20 kg difference), while on the riskier side, the 3 designs are much closer (5 to 10 kg difference). The dominating design presents a steeper Belief curve, which means that its performance is less sensitive to the inherent uncertainties present in the misison scenario.

To conclude, 3 robust designs for the low thrust mission to 1999JU3 have been found. The mission is feasible, and 80 to 120 kg of payload could be transported to the destination. The decision makers could opt for the design less sensitive to uncertainty and more efficient, but would have to accept a long transfer duration. The alternatives are more risky but have a transfer time which is 10 months shorter.

5.3 CONCLUSIONS

In this chapter, two reliable space mission designs in the frame of Evidence Theory have been tackled by the methods introduced and discussed earlier.

On the BepiColombo-like test case, a comparison of the direct and indirect methods has been presented. It confirms what was observed on the Chebyquad test cases of chapter 4.

A second application was a feasibility study of a mission to rendez-vous with an asteroid. The problem has been solved using a two-step approach. First, the BSP was used to garner knowledge about the problem, gain a first idea of the Belief curve, and identify interesting design regions. Then, the bi-objective method was run on this pruned search space to refine the results. Three alternative designs were found, showing different characteristics, performance and sensitivity to uncertainties. This should be appealing to the decision makers to allow them to take a more conscious decision.

6

FINAL REMARKS

The work presented in this thesis responds to the requirement of finding efficient ways to optimise the preliminary design of space missions. At the early stage of development, an insufficient consideration for uncertainty could lead, for instance, to a wrong decision on the feasibility of the mission. This is especially true for exploration and science missions, which are inherently subject to a large level of uncertainty. A problem formulation, combining multidisciplinary design optimisation and Evidence Theory for the crystallisation of uncertainties, has been investigated and applied to space related test cases.

In this chapter an overview of the original work carried out by the author is provided, and the main results of this thesis are summarised. On the base of the findings of this study, future works and some recommendations are given.

6.1 SUMMARY AND FINDINGS OF THE THESIS

By combining modern statistical methods to model uncertainties inherent to any preliminary phase and global search techniques for multidisciplinary design, the aim of this dissertation is to provide a methodology to generate optimised reliable design solutions that can be used in feasibility studies or during the preliminary phase of a large scale engineering development.

The first objective of this thesis was to present the uncertainty modelling and to consider the application of Evidence Theory as an alternative to Probability

Theory or *worst case* philosophy in the domain of multidisciplinary design optimisation. There is an almost universal agreement that Probability Theory is well suited to represent any form of variability when sufficient data is available. However, epistemic uncertainties due to a lack of information are acknowledged by the risk assessment and reliable engineering community as being insufficiently and incorrectly modelled by traditional theories. Recently, alternative approaches have been developed to address this issue. In chapter 2, the different typologies of uncertainties are introduced. Evidence Theory, adequate to represent both aleatory and epistemic uncertainties, is presented from a practical and application-focused angle. Even if it is a relatively recent paradigm, the theoretical foundations are well established. The applications of Evidence Theory to real life engineering cases are, however, still rare and work in this area is mostly limited to within the academic community. This can be explained by both the large computational cost associated with it and little awareness of the advantages that this approach can bring, by engineers and decision makers.

Once the mathematical framework representing uncertainties was in place, attention was given to the formulation of the Optimisation Under Uncertainty problem. After defining the concept of robust and reliable design optimisation, chapter 3 presented existing methods to tackle the problem. In particular, the *bi-objective method* introduced previously by Dr. Vasile was revisited and algorithmic improvements to make it more efficient proposed. The definition of dominance was also revisited for the particular case of robust design, and led to the *multi-belief* formulation of the problem. Dedicated algorithms have been proposed to significantly reduce the computational effort by using as much as possible available data and prematurely terminating computations that are deemed to be unnecessary. Additionally, an indirect method was proposed

to mitigate the computational cost of evaluating the Belief, a critical aspect when used as objective in an optimisation problem. The *cluster approximation method* is a sample-based technique designed such that the computational time is as much as possible linear with the problem dimensionality and complexity. Observing that epistemic uncertainties are classically represented by means of intervals led to the use of a Binary Space Partition as a dedicated clustering implementation.

Studying the efficiency of existing methods, and proposing improvements or alternative ones to solve robust design optimisation problems in the frame of Evidence Theory was the second and main objective of this thesis. To assess the presented methods, an extensive test campaign has been conducted on set analytical test cases especially designed for this work. The cost function is based on the Chebyshev polynomials, and has the particularity that its dimensionality can be modified easily. When used as a system function of the OUU, it allowed us to investigate the influence of the number of uncertain parameters and design variables. Additionally this cost function presents multiple and similar local optima. Therefore, the capability of the different methods in identifying the different optimal designs for varying levels of robustness was put to the test.

Of the studied direct methods, the multi-belief method has a better overall performance, particularly in problems with a small number of design variables and focal elements. As the complexity of the problem increases, the bi-objective method becomes more and more valuable, demonstrating the effectiveness of the algorithmic improvements introduced in this work. The BSP approximation proved to be an appropriate way to tackle OUU problems if the system function is computationally expensive to evaluate. The computational time required to approximate the optimal Belief curve and identify good designs is very

reasonable; it remains exponential with the number of focal elements, but this is confined to a very limited part of the algorithm, accounting for a fraction of the total in even the most complex test cases. The BSP method can handle problems with a very large set of focal elements in a matter of seconds.

Finally, the third motivation of this work was to highlight to the space industry the potential benefits of using a robust design methodology in general, and combined with Evidence Theory in particular. For this purpose, two examples of preliminary space mission design were used to illustrate how the proposed methodology can be applied. The context of exploration and science missions based on solar electric propulsion was chosen because of the importance of uncertainties at the early stage of the development, and the evident multidisciplinary aspect of such designs. On the BepiColombo-like test case, a comparison of the direct and indirect methods which confirmed what was observed during the Chebyquad test cases was presented. A second test case was the feasibility study of a mission to rendez-vous with an asteroid conducted as a two-step approach. First, the BSP was used to garner insight and knowledge about the problem, obtain a first idea of the Belief curve, and identify interesting design regions. The bi-objective method was then run to refine the results, converging eventually to three alternative designs. These presented different characteristics, performance and sensitivity to uncertainties, something that should be appealing to the decision makers always eager to make a conscious and educated choice.

6.2 FUTURE WORK

In the present work, techniques to solve OUU in the frame of Evidence Theory have been compared based on their performance at solving the suite of

Chebyquad test cases. It would be interesting to apply to the same set of problems reliable design optimisation approaches using alternative characterisation of uncertainties such as probability theory, possibly theory and fuzzy sets.

Further work is primarily required on the novel BSP method. A proposed modification is to limit the number of partitions along the design dimensions. Because it is an approximation method, it is illusory to identify very small design regions as the most promising ones. This should increase significantly the probability that the optimal designs are contained in the best design regions identified by the BSP. Also, greater attention could be given to the sampling process associated with the indirect method. The proposed pool function for instance could be enhanced by introducing a secondary pool threshold, thus driving the sample points away from regions where the system function returns too low as well as too high values. Also, a great variety of sample techniques exists in the literature [McKay et al., 1979, Sacks et al., 1989, Swiler et al., 2006] such as stratified sampling, Halton sampling, Hammersley sampling or Centroidal Voronoi Tesselation to name but a few. An extensive survey would be useful to identify the most suitable ones for the cluster-approximation method.

The implementation of the bi-objective method could benefit from a small modification to the way the threshold is treated by the optimiser. As well as being an objective, the threshold is a variable to the problem, and no distinction is made between the threshold and the design variables. Thus crossovers between the threshold and one of the design variables occur. The different nature of the threshold and the design variables is such that this could impair the performance of the genetic algorithm. Resolving this issue could be done, for instance, by not allowing the genetic algorithm to consider, during the creation of a new generation, the variable "threshold" for crossover.

A third area of work could be a more detailed investigation of hybrid methods, combining an indirect with a direct approach. As illustrated in the test case of a mission to an asteroid, the strength of the indirect method is to explore quickly a large search space in order to identify regions of interest. A direct method, more expensive computationally but more accurate, can be used to refine the results focusing only on the most promising design regions. On what type of problems could such a hybrid method be beneficial?, when to stop the indirect method and switch to the direct one?, are two examples of questions worth considering for future work.

Finally, a side-by-side comparison of a preliminary space mission design obtained with the margin approach, OUU in the frame of Probability Theory and in the frame of Evidence Theory would be an excellent way to show the advantages and drawbacks of these three different approaches. Increasing awareness of alternatives to the traditional margin approach to the space sector could potentially reduce cost and increase the efficiency of spacecraft.

A

SETS OF OPTIMAL DESIGNS OF THE CHEBYQUAD TEST CASES

Range of Belief Levels		Design Vector
0.3	0.3	0.173842
0.7	0.7	0.682178
1	1	0.673443

Table 24: Optimal designs for the Chebyquad test case – $n_D = 1, n_U = 1$.

Range of Belief Levels		Design Vector
0.09	0.09	0.191919
0.21	0.42	0.485028
0.49	0.51	0.44336
0.58	1	0.405518

Table 25: Optimal designs for the Chebyquad test case – $n_D = 1, n_U = 2$.

Range of Belief Levels		Design Vector
0.027	0.189	0.230804
0.21	0.216	0.260538
0.237	0.657	0.221281
0.664	1	0.250607

Table 26: Optimal designs for the Chebyquad test case – $n_D = 1, n_U = 3$.

Range of Belief Levels		Design Vector
0.0081	0.0756	0.132664
0.0819	0.3402	0.154223
0.3409	0.3483	0.085711
0.349	0.7599	0.164933
0.7606	1	0.171172

Table 27: Optimal designs for the Chebyquad test case – $n_D = 1, n_U = 4$.

Range of Belief Levels		Design Vector	
0.09	0.42	0.874115	0.372492
0.49	0.51	0.602761	0.157933
0.58	1	0.84244	0.287314

Table 28: Optimal designs for the Chebyquad test case – $n_D = 2, n_U = 2$.

Range of Belief Levels		Design Vector		
0.3	0.3	0.69472	0.345349	0.086826
0.7	0.7	0.856388	0.512679	0.257763
1	1	0.859629	0.511439	0.261674

Table 29: Optimal designs for the Chebyquad test case – $n_D = 3, n_U = 1$.

Range of Belief Levels		Design Vector		
0.09	0.42	0.757428	0.431014	0.172716
0.49	0.51	0.686636	0.334509	0.18326
0.58	0.91	0.764115	0.624332	0.212111
1	1	0.774884	0.599244	0.207528

Table 30: Optimal designs for the Chebyquad test case – $n_D = 3, n_U = 2$.

Range of Belief Levels		**Design Vector**		
0.027	0.189	0.793518	0.418702	0.135768
0.21	0.63	0.615884	0.282179	0.123454
0.637	0.657	0.550663	0.239762	0.12951
0.664	1	0.923037	0.576031	0.142915

Table 31: Optimal designs for the Chebyquad test case – $n_D = 3, n_U = 3$.

Range of Belief Levels		**Design Vector**		
0.0081	0.2646	0.699464	0.354749	0.125163
0.2664	0.3402	0.799063	0.487448	0.154594
0.3409	0.3483	0.45426	0.252255	0.132735
0.349	0.7518	0.757333	0.451786	0.105358
0.7522	0.7599	0.721529	0.441935	0.105807
0.7606	1	0.819988	0.462537	0.103319

Table 32: Optimal designs for the Chebyquad test case $n_D = 3, n_U = 4$.

Range of Belief Levels		**Design Vector**			
0.09	0.42	0.8396	0.549243	0.292853	0.116824
0.49	0.51	0.763035	0.449228	0.295807	0.097998
0.58	0.91	0.907828	0.617423	0.549481	0.166354
1	1	0.909181	0.624566	0.544437	0.166683

Table 33: Optimal designs for the Chebyquad test case – $n_D = 4, n_U = 2$.

Range of Belief Levels		**Design Vector**		
0.04	0.25	0.902088	0.62163	0.069178
0.26	0.3	0.790716	0.412628	0.079565
0.31	0.55	0.899332	0.539279	0.075084
0.56	0.75	0.824657	0.629007	0.246533
0.76	0.87	0.800269	0.548682	0.211941
0.88	0.96	0.755155	0.436385	0.172314
1	1	0.774598	0.598206	0.207061

Table 34: Optimal designs for the Chebyquad test case – $n_D = 3, n_U = 2, n_{FE} = 9$.

Range of Belief Levels		**Design Vector**		
0.01	0.2	0.945115	0.717337	0.220019
0.21	0.25	0.902853	0.623078	0.0689
0.26	0.35	0.92104	0.552536	0.068486
0.36	0.45	0.891621	0.529437	0.070581
0.46	0.55	0.90034	0.53897	0.074554
0.56	0.58	0.828584	0.621051	0.255333
0.59	0.75	0.824661	0.629153	0.246608
0.76	0.83	0.82532	0.596019	0.231908
0.84	0.87	0.800294	0.548614	0.211882
0.88	0.91	0.906597	0.515045	0.101358
0.92	0.95	0.828675	0.508124	0.180209
0.96	0.96	0.754917	0.437172	0.17337
0.97	1	0.774802	0.599312	0.2077

Table 35: Optimal designs for the Chebyquad test case -$n_D = 3, n_U = 2, n_{FE} = 16$.

Range of Belief Levels		**Design Vector**		
0.01	0.105	0.851554	0.631755	0.061778
0.1075	0.14	0.903072	0.553663	0.086327
0.1425	0.2275	0.838037	0.604303	0.045701
0.23	0.245	0.916553	0.604512	0.066377
0.2475	0.2625	0.850812	0.566599	0.051487
0.265	0.3675	0.853864	0.570215	0.051496
0.37	0.4375	0.806569	0.523843	0.061421
0.44	0.46	0.851088	0.532885	0.07782
0.4625	0.4675	0.92129	0.55206	0.068041
0.47	0.49	0.921014	0.552668	0.068637
0.4925	0.55	0.899401	0.539362	0.075122
0.5525	0.5775	0.729199	0.711689	0.289385
0.58	0.61	0.875066	0.572086	0.111317
0.6125	0.6375	0.788346	0.646746	0.264081
0.64	0.66	0.806617	0.654676	0.263607
0.6625	0.69	0.814938	0.618218	0.244094
0.6925	0.75	0.824507	0.626848	0.245544
0.7525	0.83	0.825139	0.595591	0.231732
0.8325	0.87	0.799923	0.549544	0.21286
0.8725	0.91	0.908522	0.516207	0.101321
0.9125	0.95	0.830023	0.50934	0.180105
0.9525	0.96	0.756403	0.440593	0.175134
0.9625	1	0.774429	0.599243	0.208078

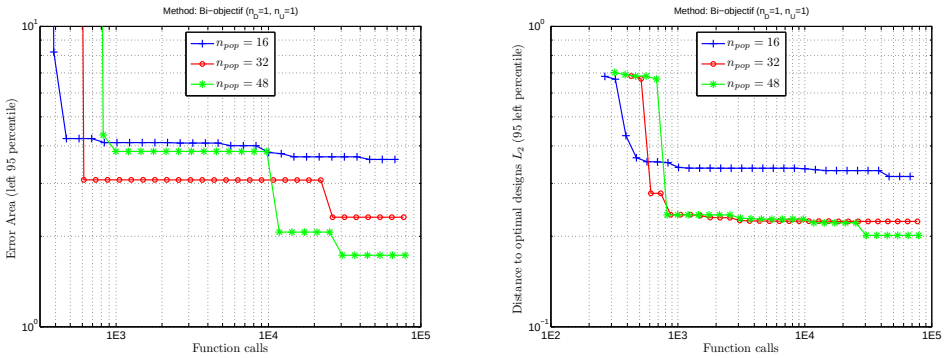
Table 36: Optimal designs for the Chebyquad test case – $n_D = 3, n_U = 2, n_{FE} = 25$.

B

ALL FIGURES OF THE CHEBYQUAD TEST CASES

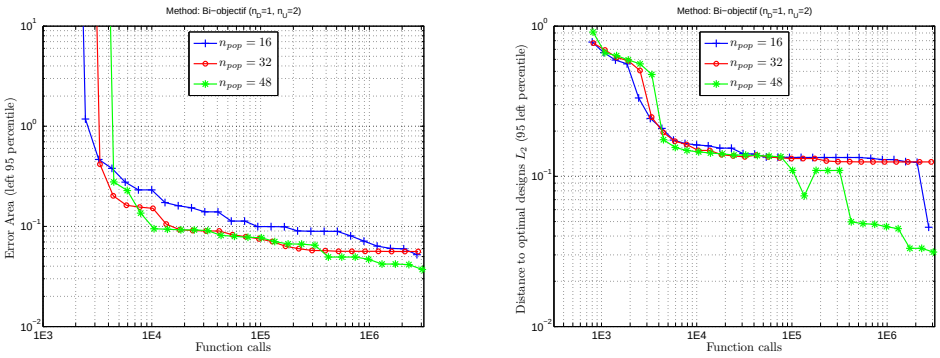
B.1 BI-OBJECTIVE METHOD

B.1.1 *Accuracy of the results*



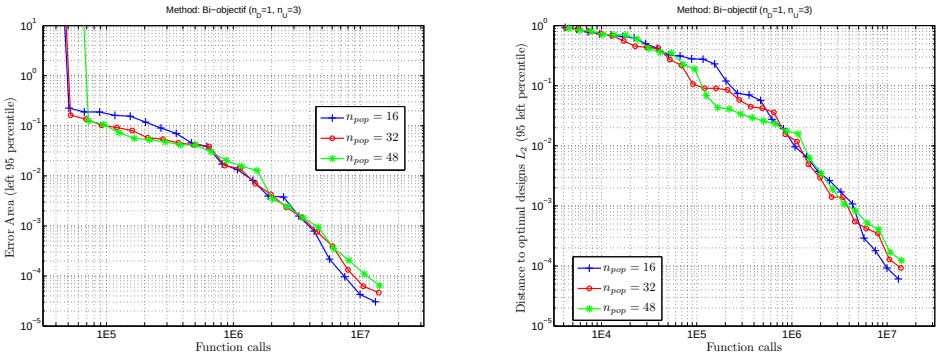
(a) (b)

Figure 56: Bi-objective, Chebyquad cases with $n_D = 1$ and $n_U = 1$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



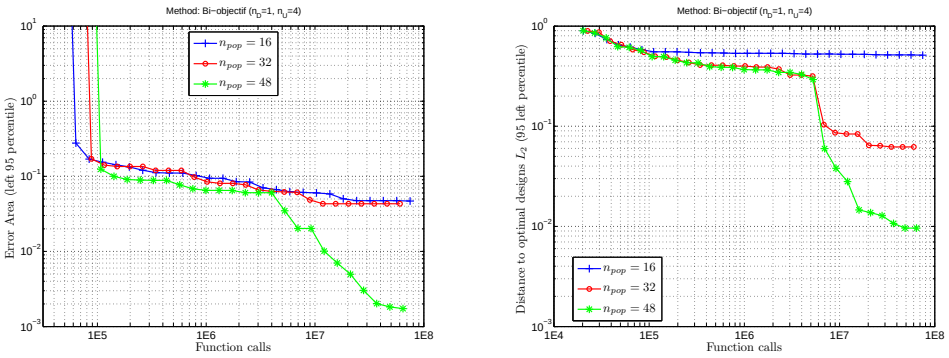
(a) (b)

Figure 57: Bi-objective, Chebyquad cases with $n_D = 1$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



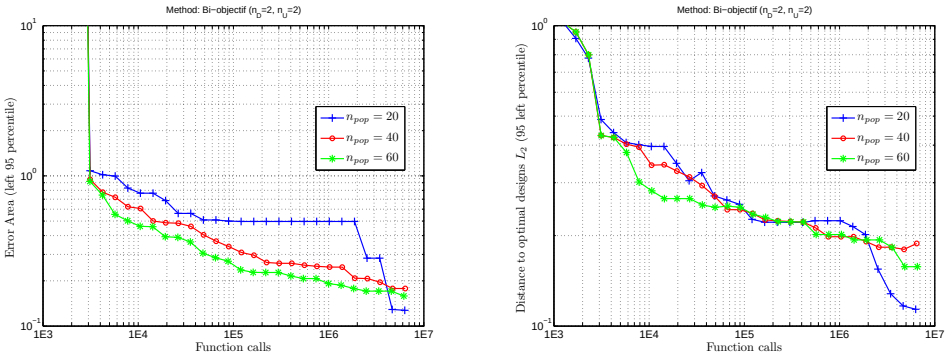
(a) (b)

Figure 58: Bi-objective, Chebyquad cases with $n_D = 1$ and $n_U = 3$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



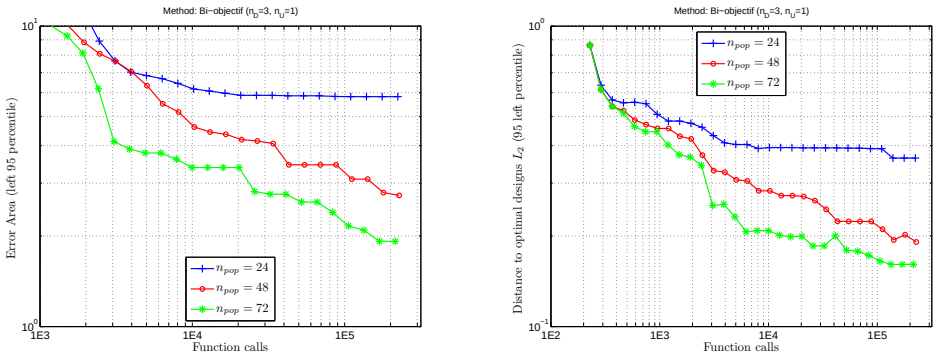
(a) (b)

Figure 59: Bi-objective, Chebyquad cases with $n_D = 1$ and $n_U = 4$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



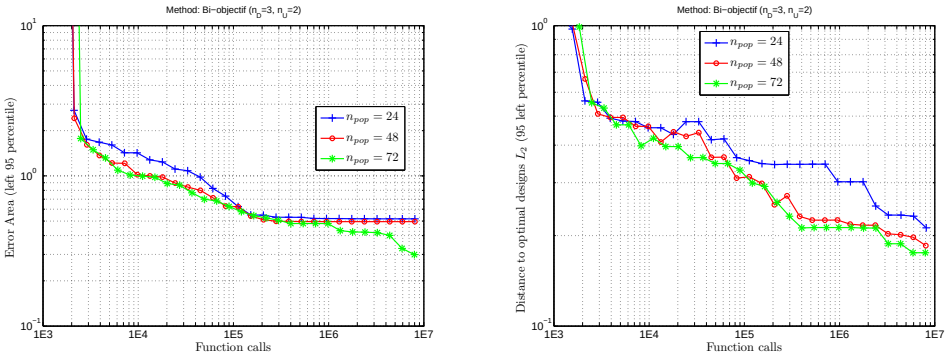
(a) (b)

Figure 60: Bi-objective, Chebyquad cases with $n_D = 2$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



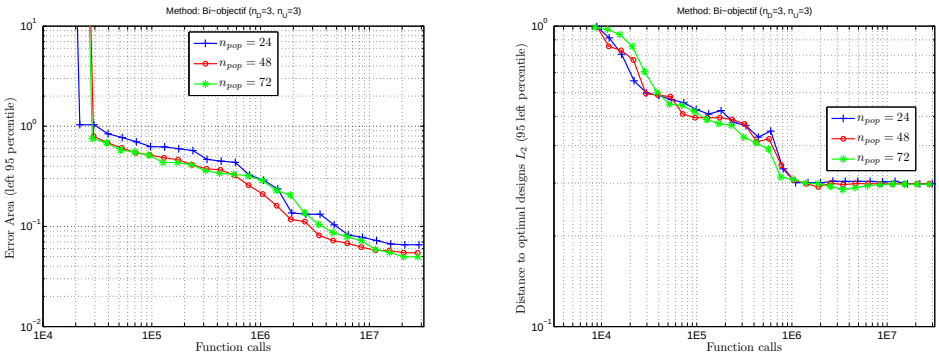
(a) (b)

Figure 61: Bi-objective, Chebyquad cases with $n_D = 3$ and $n_U = 1$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



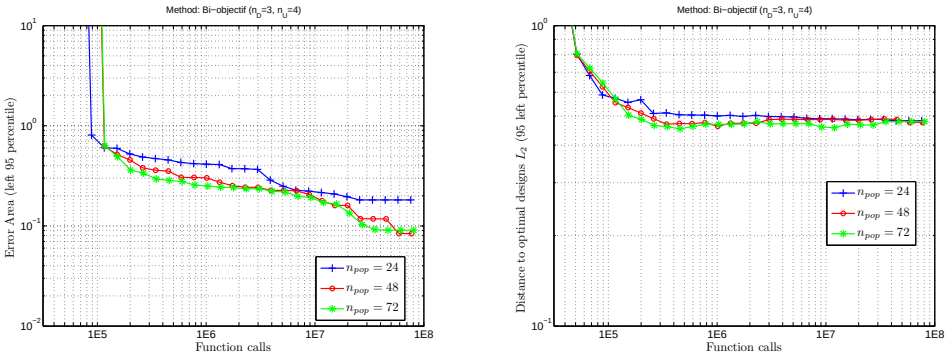
(a) (b)

Figure 62: Bi-objective, Chebyquad cases with $n_D = 3$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



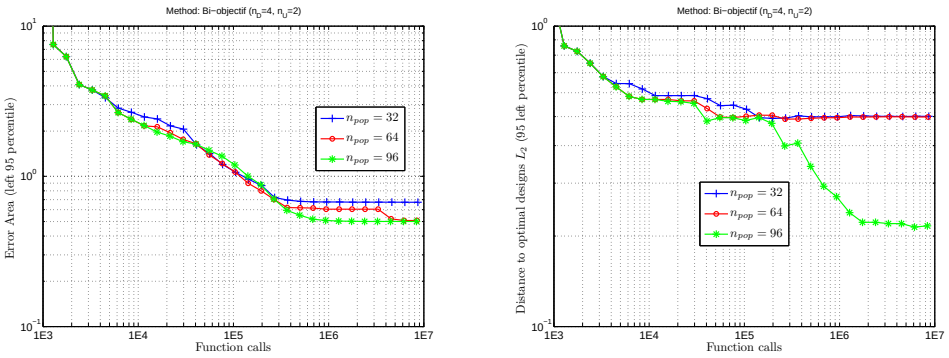
(a) (b)

Figure 63: Bi-objective, Chebyquad cases with $n_D = 3$ and $n_U = 3$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



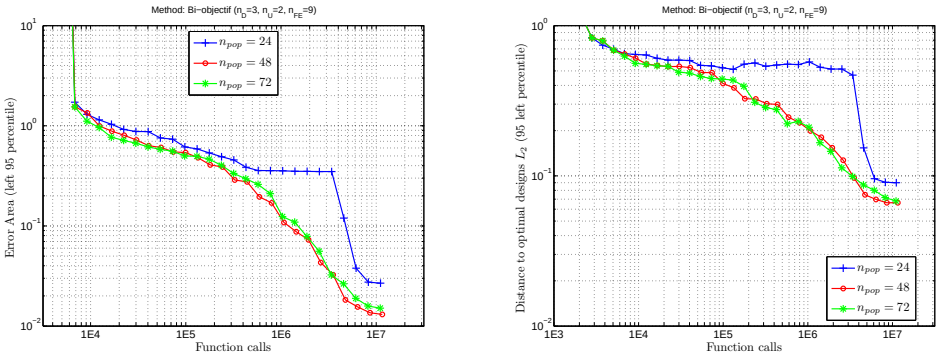
(a) (b)

Figure 64: Bi-objective, Chebyquad cases with $n_D = 3$ and $n_U = 4$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



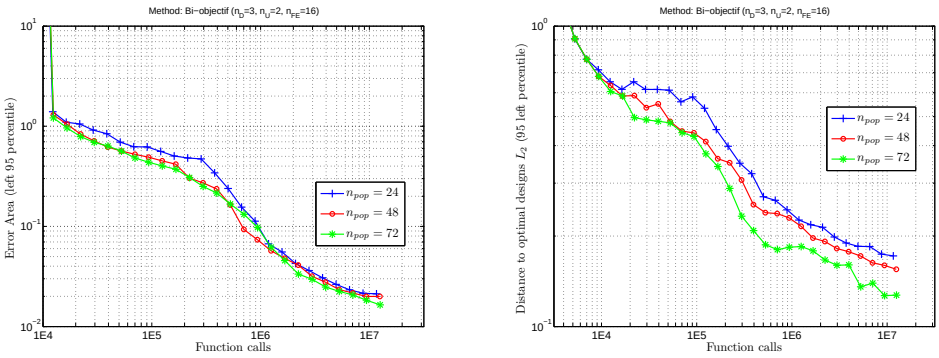
(a) (b)

Figure 65: Bi-objective, Chebyquad cases with $n_D = 4$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



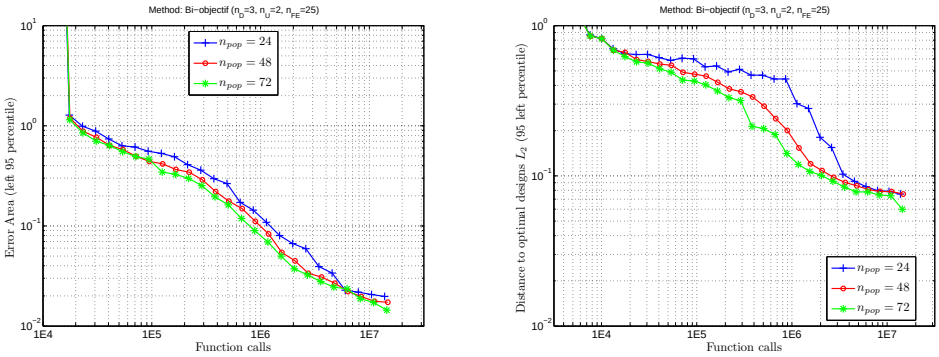
(a) (b)

Figure 66: Bi-objective, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 9$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



(a) (b)

Figure 67: Bi-objective, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 16$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.

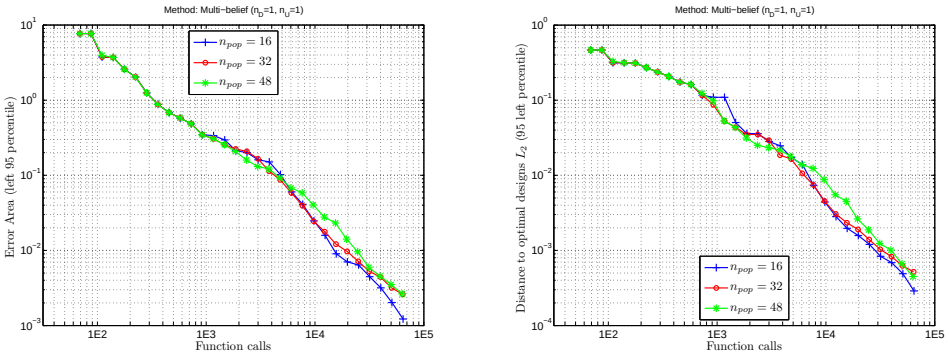


(a) (b)

Figure 68: Bi-objective, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 25$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.

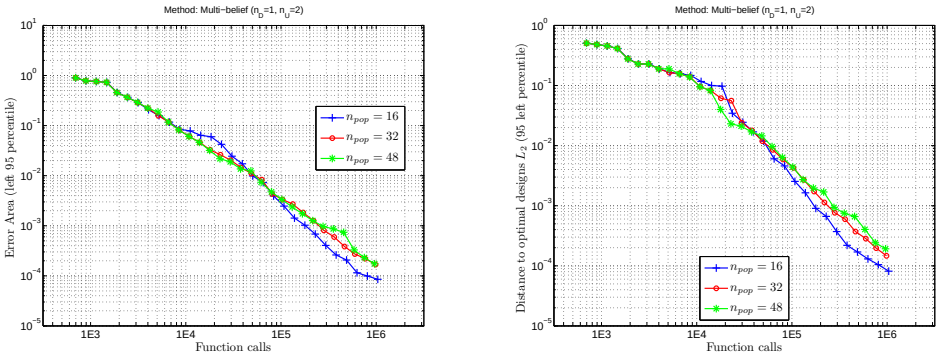
B.2 MULTI-BELIEF METHOD

B.2.1 *Accuracy of the results*



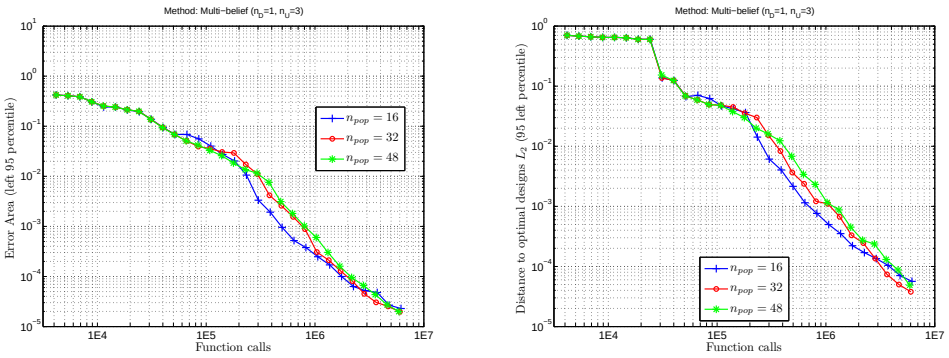
(a) (b)

Figure 69: Multi-belief, Chebyquad cases with $n_D = 1$ and $n_U = 1$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



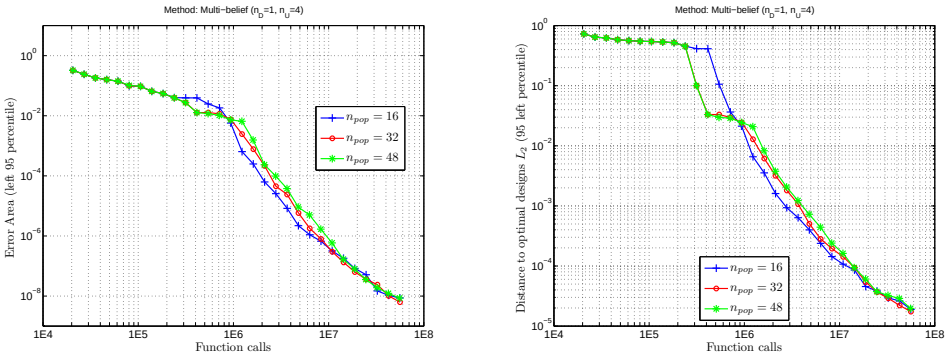
(a) (b)

Figure 70: Multi-belief, Chebyquad cases with $n_D = 1$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



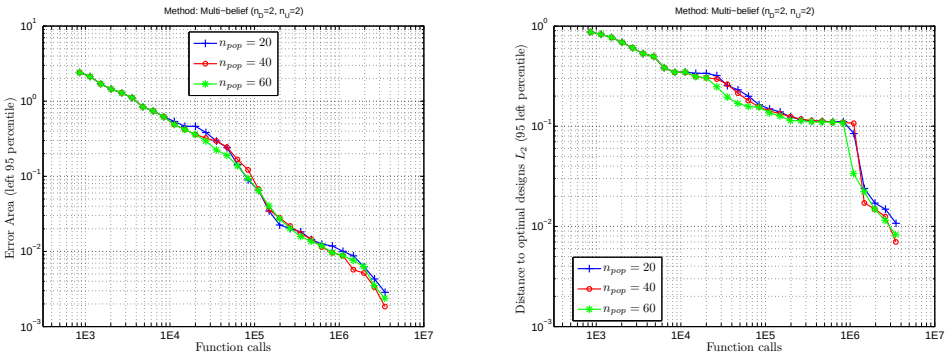
(a) (b)

Figure 71: Multi-belief, Chebyquad cases with $n_D = 1$ and $n_U = 3$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



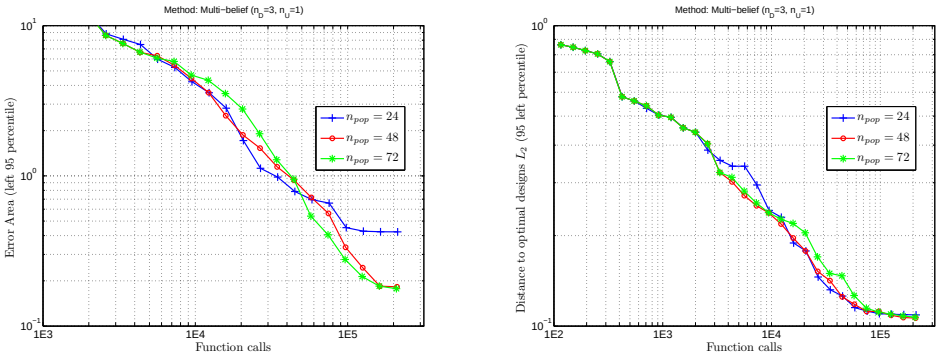
(a) (b)

Figure 72: Multi-belief, Chebyquad cases with $n_D = 1$ and $n_U = 4$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



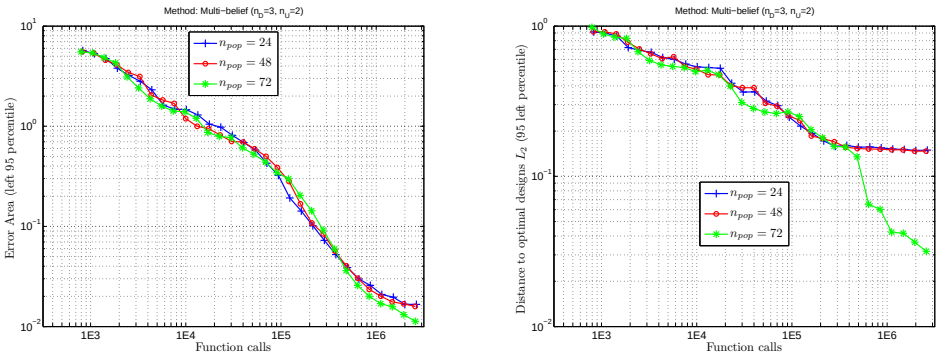
(a) (b)

Figure 73: Multi-belief, Chebyquad cases with $n_D = 2$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



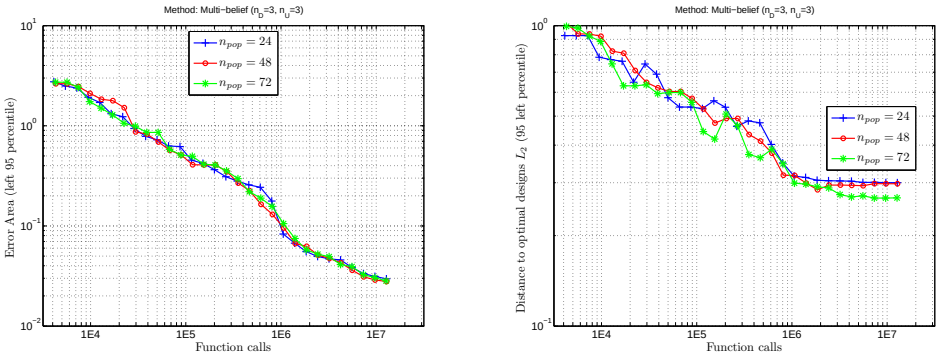
(a) (b)

Figure 74: Multi-belief, Chebyquad cases with $n_D = 3$ and $n_U = 1$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



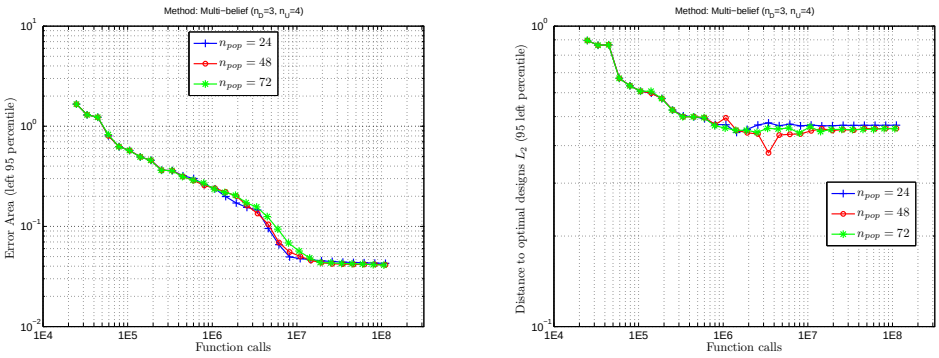
(a) (b)

Figure 75: Multi-belief, Chebyquad cases with $n_D = 3$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



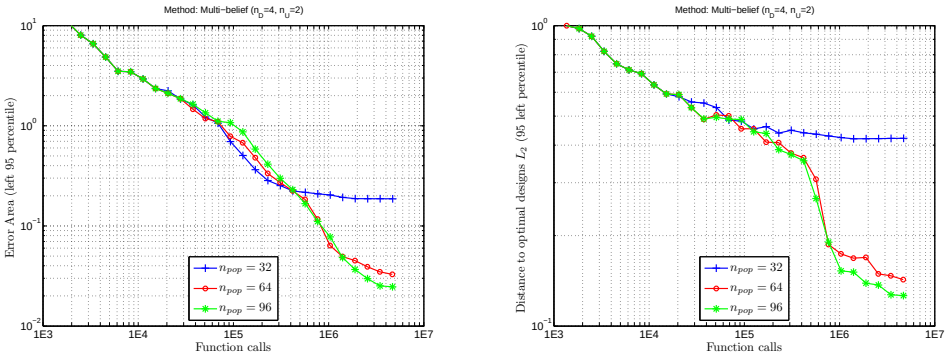
(a) (b)

Figure 76: Multi-belief, Chebyquad cases with $n_D = 3$ and $n_U = 3$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



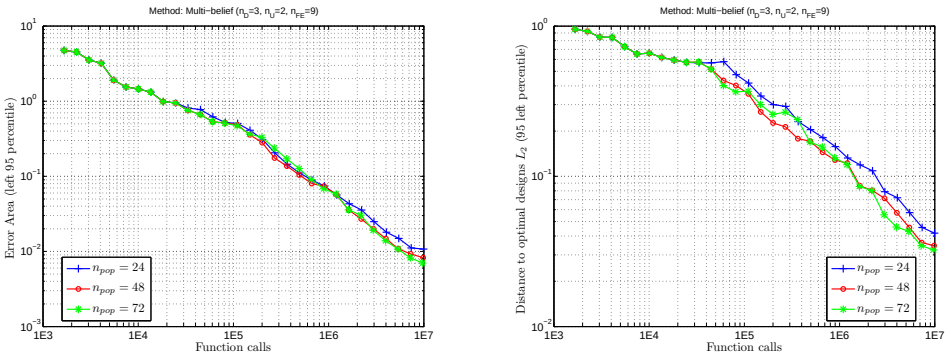
(a) (b)

Figure 77: Multi-belief, Chebyquad cases with $n_D = 3$ and $n_U = 4$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



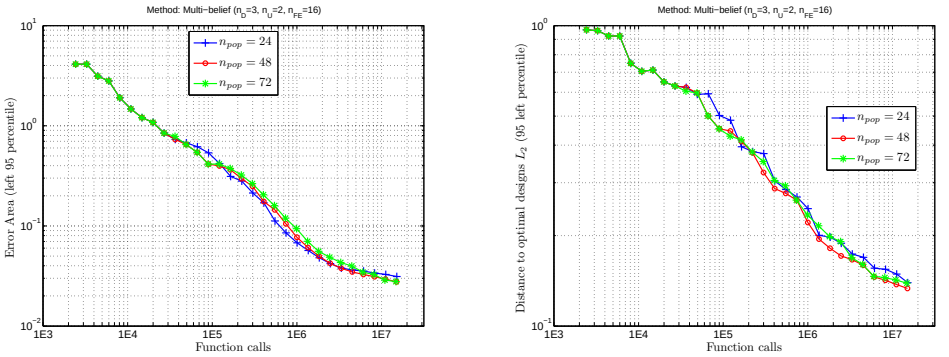
(a) (b)

Figure 78: Multi-belief, Chebyquad cases with $n_D = 4$ and $n_U = 2$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



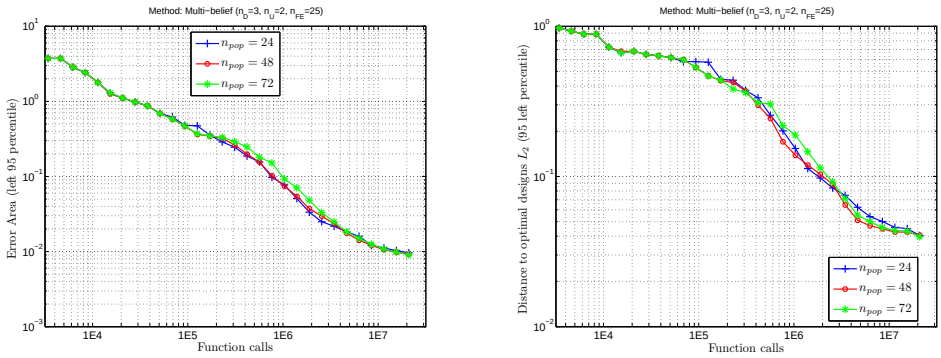
(a) (b)

Figure 79: Multi-belief, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 9$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.



(a) (b)

Figure 80: Multi-belief, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 16$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.

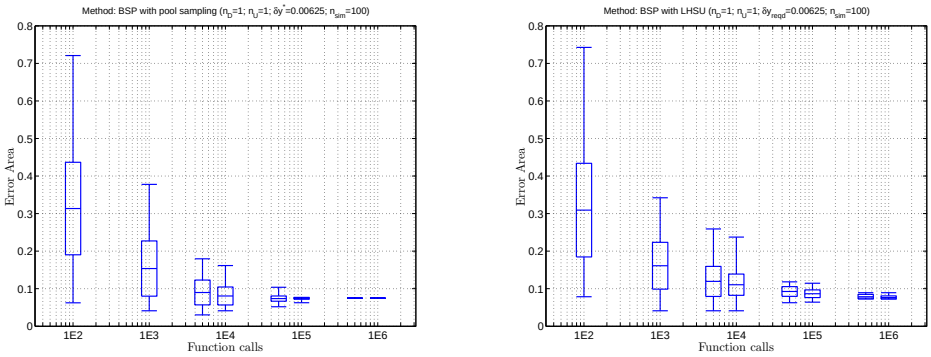


(a) (b)

Figure 81: Multi-belief, Chebyquad cases with $n_D = 3$, $n_U = 2$ and $n_{FE} = 25$, 95 percentile value of: (a) error area, and (b) distance to optimal designs.

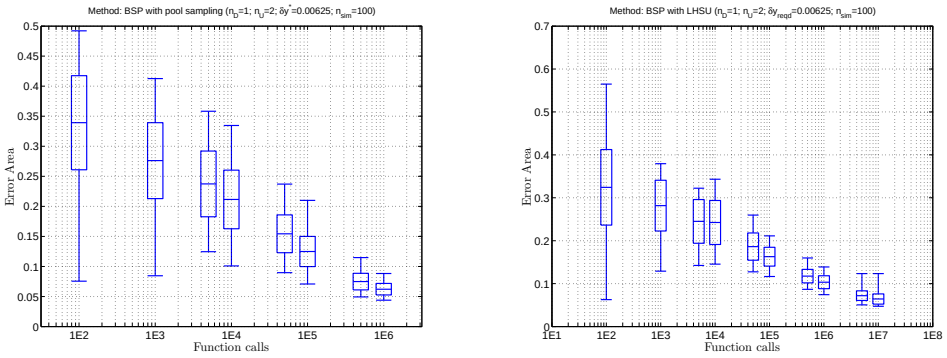
B.3 BSP METHOD

B.3.1 *Box plots of the error area*



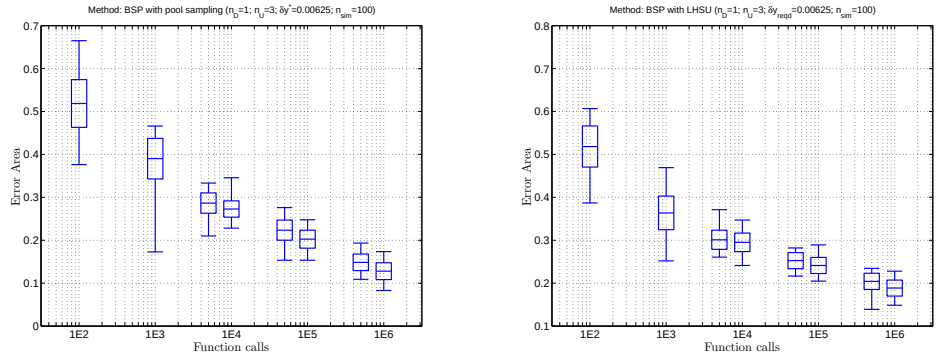
(a) (b)

Figure 82: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 1$ and $n_U = 1$: (a) with pool sampling, and (b) with LHSU.



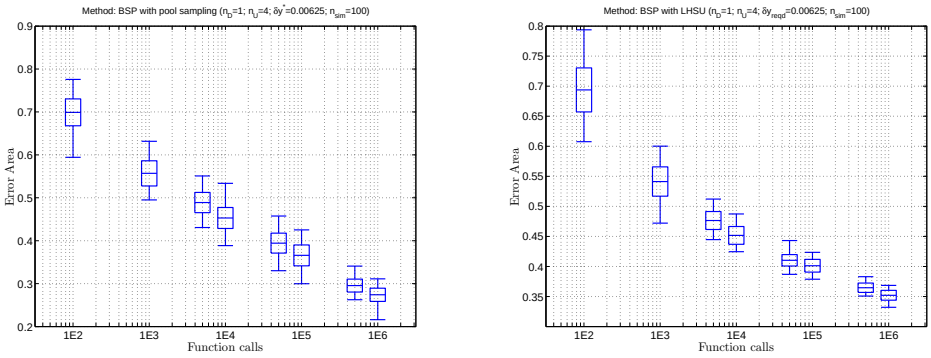
(a) (b)

Figure 83: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 1$ and $n_U = 2$: (a) with pool sampling, and (b) with LHSU.



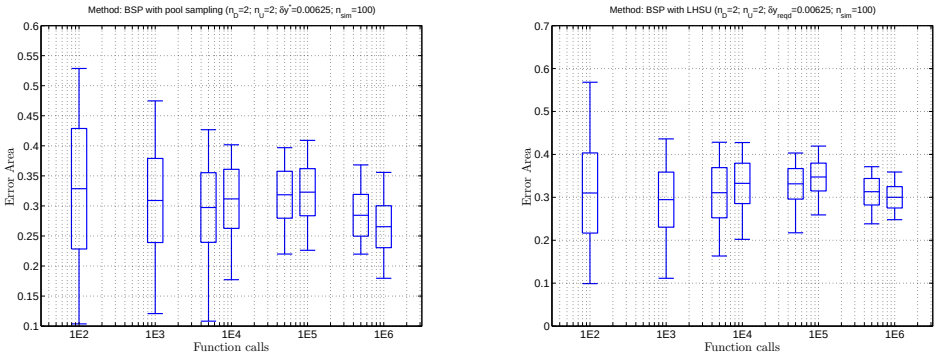
(a) (b)

Figure 84: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 1$ and $n_U = 3$: (a) with pool sampling, and (b) with LHSU.



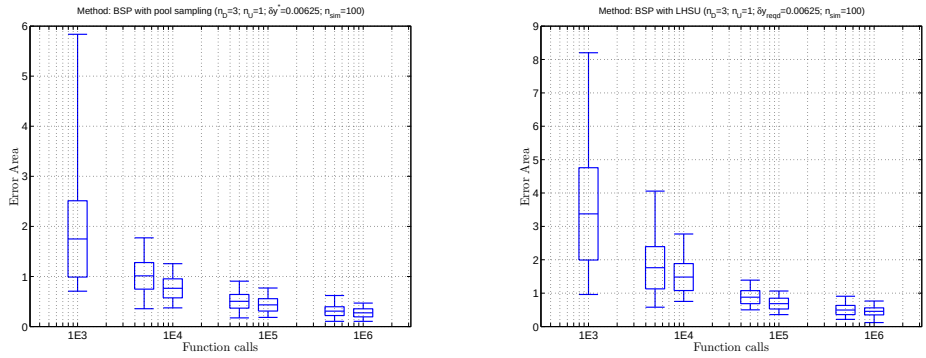
(a) (b)

Figure 85: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 1$ and $n_U = 4$: (a) with pool sampling, and (b) with LHSU.



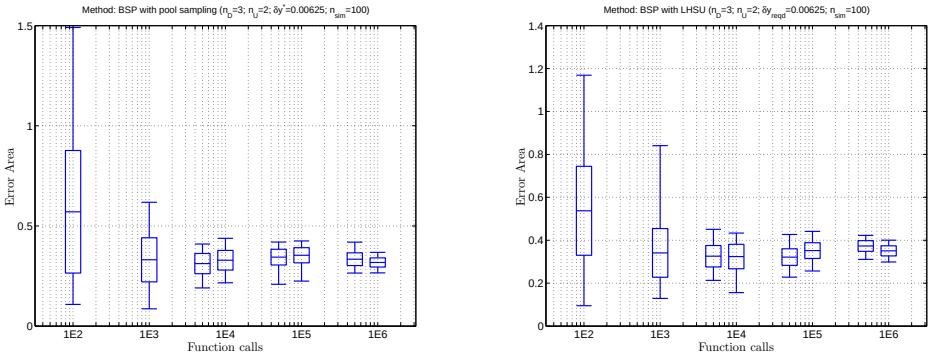
(a) (b)

Figure 86: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 2$ and $n_U = 2$: (a) with pool sampling, and (b) with LHSU.



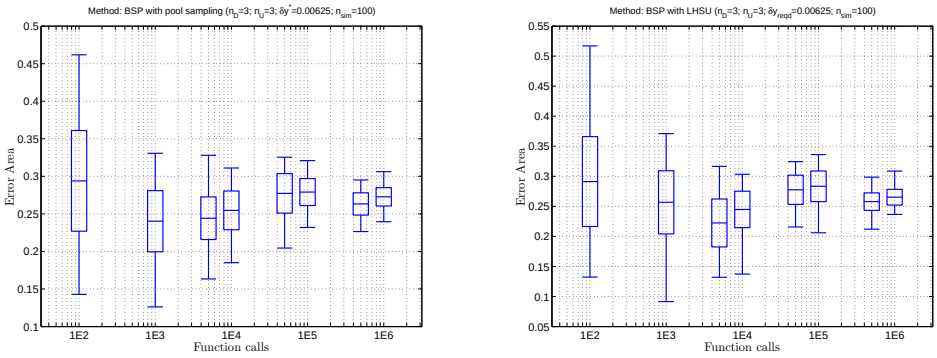
(a) (b)

Figure 87: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$ and $n_U = 1$: (a) with pool sampling, and (b) with LHSU.



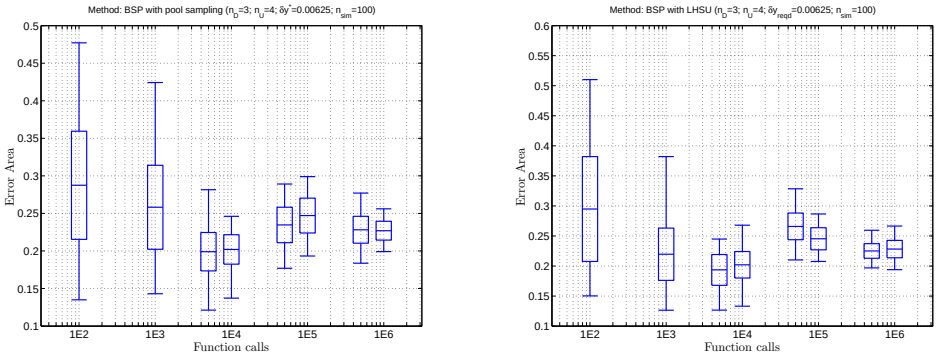
(a) (b)

Figure 88: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$ and $n_U = 2$: (a) with pool sampling, and (b) with LHSU.



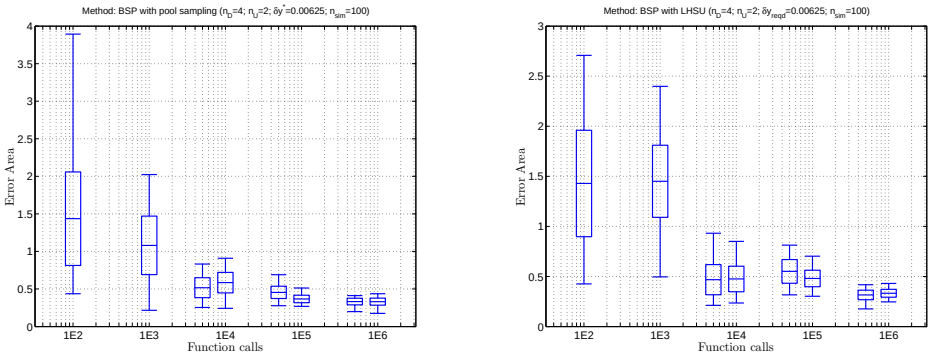
(a) (b)

Figure 89: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$ and $n_U = 3$: (a) with pool sampling, and (b) with LHSU.



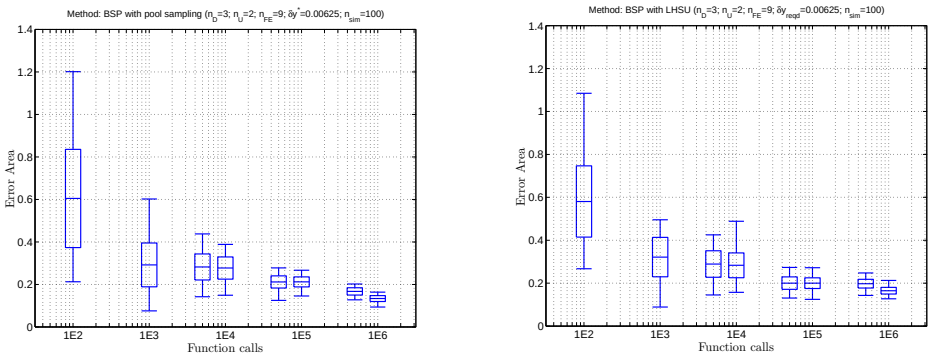
(a) (b)

Figure 90: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$ and $n_U = 4$: (a) with pool sampling, and (b) with LHSU.



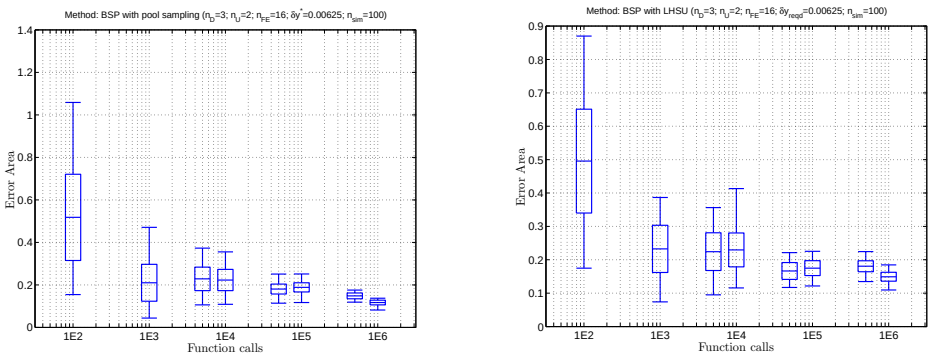
(a) (b)

Figure 91: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 4$ and $n_U = 2$: (a) with pool sampling, and (b) with LHSU.



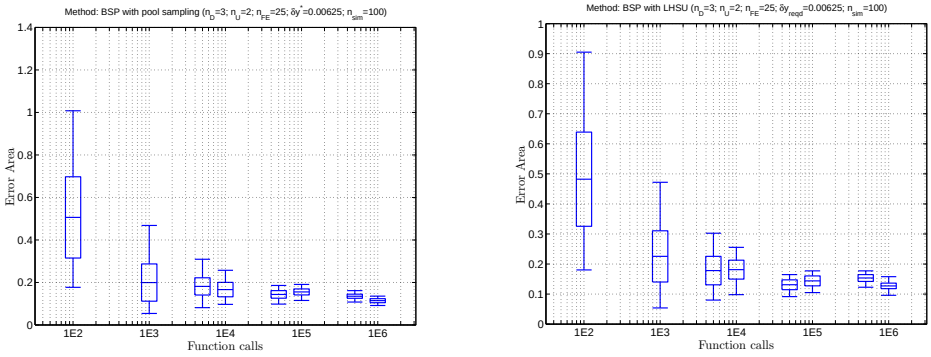
(a) (b)

Figure 92: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$, $n_U = 2$ and $n_{FE} = 9$: (a) with pool sampling, and (b) with LHSU.



(a) (b)

Figure 93: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$, $n_U = 2$ and $n_{FE} = 16$: (a) with pool sampling, and (b) with LHSU.



(a) (b)

Figure 94: BSP, with threshold step $y^* = 0.00625$, on the Chebyquad case with $n_D = 3$, $n_U = 2$ and $n_{FE} = 25$: (a) with pool sampling, and (b) with LHSU.

C

DEEP SPACE ΔV SAMPLES POINTS

Deep space ΔV $[\mathrm{m.s^{-1}}]$	**Power to be generated at 1AU** P_{1AU} [kW]	**Maximum thrust** T_{max} [mN]									
	210	230	250	270	290	310	330	350	370	400
4.20		1985.70	2022.70	2028.49	2030.16	2030.79	2031.14	2031.30	2030.39	2031.91
4.25	1996.42	2022.88	2032.09	2036.69	2039.60	2039.71	2039.85	2040.07	2039.80	2040.44
4.30	2003.31	2029.07	2037.71	2041.95	2045.20	2045.48	2045.49	2045.65	2045.63	2045.84
4.35	2006.39	2032.64	2041.38	2045.45	2048.88	2049.58	2049.59	2049.49	2049.69	2049.79
4.50	2010.35	2039.18	2048.56	2052.39	2056.17	2058.49	2058.48	2058.35	2058.35	2058.43
4.65	2012.84	2043.27	2053.14	2057.09	2061.01	2064.23	2064.80	2064.68	2064.67	2064.62
4.80	2014.92	2045.11	2056.47	2060.51	2064.61	2067.94	2069.67	2069.59	2069.60	2069.54
4.95	2016.63	2046.66	2059.15	2064.65	2067.52	2070.97	2073.24	2073.69	2073.68	2073.61
5.10	2017.95	2048.08	2060.20	2066.92	2069.81	2073.41	2075.81	2077.13	2077.12	2077.10
5.25	2019.04	2049.48	2061.06	2068.92	2071.84	2075.54	2078.04	2079.72	2080.13	2080.00
5.40	2019.90	2050.72	2062.24	2069.58	2073.69	2077.36	2079.95	2081.73	2082.72	2082.61
5.55	2021.06	2051.83	2063.44	2070.06	2075.19	2078.87	2081.52	2083.46	2084.64	2084.85
5.70	2021.37	2052.70	2064.52	2070.97	2075.67	2080.07	2082.96	2084.93	2086.17	2086.78
5.85	2021.51	2053.90	2065.35	2071.86	2075.83	2081.05	2084.17	2086.17	2087.49	2088.45
6.00	2022.10	2054.61	2066.38	2072.63	2076.34	2081.56	2085.21	2087.25	2088.65	2089.73
6.15	2022.64	2054.83	2067.28	2073.25	2076.97	2081.47	2086.13	2088.21	2089.62	2090.80
6.30	2023.04	2054.87	2068.31	2074.11	2077.57	2081.91	2086.35	2089.03	2090.48	2091.71
6.45	2023.41	2055.55	2068.69	2074.88	2078.17	2082.35	2085.95	2089.77	2091.25	2092.60

Table 37: Sample points of the deep space ΔV over a grid of P_{1AU} and T_{max} for BepiColombo.

BIBLIOGRAPHY

[Agarwal et al., 2003] Agarwal, H., Renaud, J. E., and Preston, E. L. 2003. Trust region managed reliability based design optimization using evidence theory. Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 5:3449 – 3463. (Cited on pages xiii, 4, 11, 21, 25, 33, 42, and 128.)

[Agarwal et al., 2004] Agarwal, H., Renaud, J. E., Preston, E. L., and Padmanabhan, D. 2004. Uncertainty quantification using evidence theory in multidisciplinary design optimization. Reliability Engineering and System Safety, 85(1-3):281 – 294. (Cited on page 42.)

[Bae et al., 2002] Bae, H.-R., Grandhi, R. V., and Canfield, R. A. 2002. Uncertainty quantification of structural response using evidence theory. Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 4:2135 – 2145. (Cited on pages 4, 13, 25, and 33.)

[Bailey and Cowles, 1987] Bailey, T. and Cowles, J. 1987. A convex hull inclusion test. IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(2):312–316. (Cited on page 55.)

[Barnett, 1981] Barnett, J. A. 1981. Computational methods for a mathematical theory of evidence. In: Proceedings of the 7th international joint conference on Artificial intelligence - Volume 2, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pages 868–875. (Cited on page 5.)

[Bartels et al., 1987] Bartels, R. H., Beatty, J. C., and Barsky, B. A. 1987. An introduction to splines for use in computer graphics & geometric modeling. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited on page 130.)

[Bates and Wynn, 2004] Bates, R. A. and Wynn, H. P. 2004. Modelling feasible design regions using lattice-based kernel methods. Quality and Reliability Engineering International, 20(2):135 – 142. Computer experiments;Kriging models;Input constraints;Hilbert basis;Number-theoretic methods;Lattices;Lattice-based kernel methods;. (Cited on page 55.)

[Bauer, 1997] Bauer, M. 1997. Approximation algorithms and decision making in the Dempster-Shafer Theory of Evidence - an empirical study. International Journal of Approximate Reasoning, 17(2-3):217 – 237. (Cited on pages 30 and 31.)

[Benkhoff et al., 2010] Benkhoff, J., Van Casteren, J., Hayakawa, H., Fujimoto, M., Laakso, H., Novara, M., Ferri, P., Middleton, H. R., and Ziethe, R. 2010. BepiColombo - comprehensive exploration of mercury: Mission overview and science goals. Planetary and Space Science, 58(1-2):2–20. (Cited on page 120.)

[Bunday, 1984] Bunday, B. D. 1984. Basic Linear programming. Hodder Arnold. (Cited on page 55.)

[Burrus and Lesage, 2003] Burrus, N. and Lesage, D. 2003. Theory of Evidence. Technical Report 0307, Laboratoire de Recherche et Développement de l'Epita. (Cited on pages 28 and 32.)

[Campins et al., 2009] Campins, H., Emery, J. P., Kelley, M., Fernández, Y., Licandro, J., Delbó, M., Barucci, A., and Dotto, E. 2009. Spitzer observations of spacecraft target 162173 (1999 ju3). Astronomy and Astrophysics, 503(2):L17–L20. (Cited on page 141.)

[Chen et al., 2005] Chen, L.-Z., Shi, W.-K., Deng, Y., and Zhu, Z.-F. 2005. New fusion approach based on distance of evidences. Journal of Zhejiang University: Science, 6 A(5):476 – 482. (Cited on page 17.)

[Chen et al., 1996] Chen, W., Allen, J. K., Tsui, K.-L., and Mistree, F. 1996. A procedure for robust design: Minimizing variations caused by noise factors and control factors. Journal of Mechanical Design, 118(4):478–485. (Cited on page 4.)

[Chen et al., 1998] Chen, W., Wiecek, M. M., and Zhang, J. 1998. Quality utility - a compromise programming approach to robust design. Technical report, Department of Mechanical Engineering, University of Illinois, Chicago. (Cited on page 4.)

[Clemen and Winkler, 1999] Clemen, R. T. and Winkler, R. L. 1999. Combining probability distributions from experts in risk analysis. Risk Analysis, 19(2):187–203. (Cited on page 27.)

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619. (Cited on page 53.)

[Cressie, 1990] Cressie, N. 1990. The origins of kriging. Mathematical Geology, 22:239–252. 10.1007/BF00889887. (Cited on page 130.)

[Croisard et al., 2010] Croisard, N., Vasile, M., Kemble, S., and Radice, G. 2010. Preliminary space mission design under uncertainty. Acta Astronautica, 66(5-6):654–664. (Cited on page 33.)

[Dantzig, 1965] Dantzig, G. B. 1965. Linear Programming and Extensions. Princeton University Press. (Cited on page 55.)

[Das, 2000] Das, I. 2000. Robustness optimization for constrained nonlinear programming problems. Engineering Optimization, 32(5):585–618. (Cited on page 4.)

[Davis, 2002] Davis, J. C. 2002. Statistics and Data Analysis in Geology, volume 83. John Wiley & Sons. (Cited on page 130.)

[Deb and Agrawal, 1995] Deb, K. and Agrawal, R. B. 1995. Simulated binary crossover for continuous search space. Complex Systems, 9(2):115–148. (Cited on page 78.)

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182 – 197. Multiobjective optimization;. (Cited on pages 44 and 78.)

[Deb et al., 2007] Deb, K., Sindhya, K., and Okabe, T. 2007. Self-adaptive simulated binary crossover for real-parameter optimization. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation, New York, NY, USA. ACM, pages 1187–1194. (Cited on page 78.)

[Dempster, 1966] Dempster, A. P. 1966. New methods for reasoning towards posterior distributions based on sample data. The Annals of Mathematical Statistics, 37(2):355–374. (Cited on page 5.)

[Dempster, 1967] Dempster, A. P. 1967. Upper and lower probabilities induced by a multivalued mapping. The Annals of Mathematical Statistics, 38:325–339. (Cited on pages 5 and 12.)

[Denoeux, 2000] Denoeux, T. 2000. Inner and outer clustering approximations of belief structures. In: Proceedings of IPMU'2000, Madrid. 1:125–132. (Cited on page 31.)

[Denoeux, 2001] Denoeux, T. 2001. Inner and outer approximation of belief structures using a hierarchical clustering approach. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems, 9(4):437 – 460. (Cited on page 31.)

[Du and Chen, 2002] Du, X. and Chen, W. 2002. Efficient uncertainty analysis methods for multidisciplinary robust design. AIAA Journal, 40(3):545 – 552. (Cited on page 4.)

[Du et al., 2000] Du, X., Wang, Y., and Chen, W. 2000. Methods for robust multidisciplinary design. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, 41st, Atlanta, GA. (1785). (Cited on page 4.)

[Dubois and Prade, 1989] Dubois, D. and Prade, H. 1989. Fuzzy sets, probability and measurement. European Journal of Operational Research, 40(2):135 – 154. (Cited on pages 5 and 12.)

[Dubois and Prade, 1992] Dubois, D. and Prade, H. 1992. On the combination of evidence in various mathematical frameworks. In: flamm, J. and Luisi, T. (Editors), Reliability Data Collection and Analysis, pages 213 – 241. J. Famm and T. Luisi, Brussels. (Cited on pages 16 and 17.)

[EADS Astrium, 2011] EADS Astrium 2011. BepiColombo - Mission to Mercury. `http://www.astrium.eads.net/en/programme/bepicolombo-science-mercury.html`. Last retrieved 30/12/2011. (Cited on page 120.)

[Eberly, 2000] Eberly, D. H. 2000. 3D game engine design: a practical approach to real-time computer graphics. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited on page 58.)

[Eldred et al., 2002] Eldred, M. S., Giunta, A. A., Wojtkiewicz, S. F., and Trucano, T. G. 2002. Formulations for surrogate-based optimization under uncertainty. In: AIAA / ISSMO Symposium on Multisciplinary Analysis and Optimization. 5585(September):1–12. (Cited on pages 42 and 128.)

[Engemann et al., 1996] Engemann, K. J., Miller, H. E., and Yager, R. R. 1996. Decision making with belief structures: an application in risk management. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 4:1–25. (Cited on page 33.)

[ESA, 2011] ESA 2011. BepiColombo in depth. `http://sci.esa.int/bepicolombo`. Last retrieved 28/12/2011. (Cited on pages 120 and 121.)

[Ferson et al., 2004] Ferson, S., Nelsen, R. B., Hajagos, J., Berleant, D. J., Zhang, J., Tucker, W. T., Ginzburg, L. R., and Oberkampf, W. L. 2004. Dependence in probabilistic modeling, Dempster-Shafer theory, and probability bounds analysis. Technical Report SAND2004-3072, Sandia National Laboratories. (Cited on page 15.)

[Fletcher, 1965] Fletcher, R. 1965. Function minimization without evaluating derivatives–a review. The Computer Journal, 8(1):33–41. (Cited on pages 62 and 64.)

[Fuchs et al., 1980] Fuchs, H., Kedem, Z. M., and Naylor, B. F. 1980. On visible surface generation by a priori tree structures. In: Computer Graphics. 14(3):124–133. (Cited on page 56.)

[Grady, 2006] Grady, J. O. 2006. System requirements analysis. Elsevier Academic Press. (Cited on page 3.)

[Hasegawa et al., 2008] Hasegawa, S., Müller, T. G., Kawakami, K., Kasuga, T., Wada, T., Ita, Y., Takato, N., Terada, H., Fujiyoshi, T., and Abe, M. 2008. Albedo, Size, and Surface Characteristics of Hayabusa-2 Sample-Return Target 162173 1999 JU3 from AKARI. Publications of the Astronomical Society of Japan, 60(SP2):S399–S405. (Cited on page 141.)

[Hawe and Sykulski, 2007] Hawe, G. and Sykulski, J. 2007. Considerations of accuracy and uncertainty with kriging surrogate models in single-objective electromagnetic design optimisation. IET Science, Measurement and Technology, 1(1):37 – 47. (Cited on page 128.)

[Hayes, 2003] Hayes, B. 2003. A lucid interval. American Scientist, 91(6):484–488. (Cited on pages 5 and 12.)

[Helton, 1997] Helton, J. C. 1997. Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty. Journal of Statistical Computation and Simulation, 57:3–76. (Cited on page 11.)

[Helton et al., 2006] Helton, J. C., Johnson, J., Oberkampf, W. L., and Sallaberry, C. 2006. Sensitivity analysis in conjunction with evidence theory representations of epistemic uncertainty. Reliability Engineering and System Safety, 91(10-11):1414 – 34. evidence theory;epistemic uncertainty;sampling-based sensitivity analysis;exploratory analysis;stepwise analysis;cumulative belief functions;. (Cited on page 49.)

[Helton et al., 2007] Helton, J. C., Johnson, J., Oberkampf, W. L., and Storlie, C. 2007. A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory. Computer Methods in Applied Mechanics and Engineering, 196(37-40 SPEC ISS):3980 – 3998. (Cited on pages 11, 25, 31, and 49.)

[Hoare, 1962] Hoare, C. A. R. 1962. Quicksort. The Computer Journal, 5(1):10–16. (Cited on page 67.)

[Hoffman and Hammonds, 1994] 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):707–712. (Cited on pages 4 and 25.)

[Inagaki, 1991] Inagaki, T. 1991. Interdependence between safety-control policy and multiple-sensor schemes via Dempster-Shafer theory. IEEE Transactions on Reliability, 40:182 – 188. (Cited on page 16.)

[J. Holmberg et al., 1989] J. Holmberg, J., Silvennoinen, P., and Vira, J. 1989. Application of the Dempster-Shafer theory of evidence for accident probability estimates. Reliability Engineering and System Safety, 26(1):47–58. (Cited on page 33.)

[JAXA, 2011] JAXA 2011. Asteroid explorer Hayabusa-2. `http://www.jspec.jaxa.jp/e/activity/hayabusa2.html`. Last retrieved 28/12/2011. (Cited on page 141.)

[Jehn et al., 2004] Jehn, R., Campagnola, S., Garcia, D., and Kemble, S. 2004. Low-thrust approach and gravitational capture at Mercury. In: 18th International Symposium on Space Flight Dynamics. 548:487. (Cited on pages 120 and 122.)

[Jones, 2001] Jones, D. R. 2001. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization, 21(4):345 – 383. (Cited on pages 128 and 129.)

[Jousselme et al., 2002] Jousselme, A.-L., Grenier, D., and Bosse, E. 2002. Analyzing approximation algorithms in the theory of evidence. In: Dasarathy, B. V. (Editor), Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. 4731:65–74. (Cited on page 5.)

[Kemble, 2006] Kemble, S. 2006. Interplanetary Mission Analysis and Design. Springer Praxis Books. (Cited on page 123.)

[Keynes, 1921] Keynes, J. M. 1921. A Treatise on Probability. Macmillan, London. (Cited on page 26.)

[Klir and Smith, 2001] Klir, G. J. and Smith, R. M. 2001. On measuring uncertainty and uncertainty-based information: Recent developments. Annals of Mathematics and Artificial Intelligence, 32(1-4):5–33. (Cited on page 12.)

[Kreinovich et al., 2006] Kreinovich, V., Xiang, G., Starks, S. A., Longpr, L., Ceberio, M., Araiza, R., Beck, J., Kandathi, R., Nayak, A., Torres, R., and Hajagos, J. G. 2006. Towards combining probabilistic and interval uncertainty in engineering calculations: Algorithms for computing statistics under interval uncertainty, and their computational complexity. Reliable Computing, 12:471–501. (Cited on pages 5 and 12.)

[Le Duy et al., 2010] Le Duy, T. D., Vasseur, D., Dieulle, L., Bérenguer, C., and Couplet, M. 2010. Representation of parameter uncertainty with Evidence Theory in probabilistic risk assessment. In: Workshop on the Theory of Belief Functions, Brest, France. (Cited on pages 26 and 33.)

[Le Hégarat-Mascle et al., 1998] Le Hégarat-Mascle, S., Bloch, I., and Vidal-Madjar, D. 1998. Introduction to neighborhood information in the Evidence Theory and application to fusion of radar and optical images with partial clouds cover. Pattern Recognition, 31(11):1811 – 1823. (Cited on page 32.)

[Li et al., 2010] Li, J., Luo, S., and Jin, J. S. 2010. Sensor data fusion for accurate cloud presence prediction using Dempster-Shafer Evidence Theory. Sensors, 10(10):9384–9396. (Cited on page 33.)

[Limbourg, 2005] Limbourg, P. 2005. Multi-objective optimization of problems with epistemic uncertainty. In: EMO '05 Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico. pages 413–427. (Cited on pages 18 and 45.)

[Lophaven et al., 2002] Lophaven, S. N., Nielsen, H. B., and Sondergaard, J. 2002. DACE: a MatLab kriging toolbox. Technical Report IMM-TR-2002-12, Technical University of Denmark. (Cited on pages 128 and 129.)

[McKay et al., 1979] McKay, M., Beckman, R., and Conover, W. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2):239–245. (Cited on pages 80 and 166.)

[Mirkin, 2005] Mirkin, B. 2005. Clustering for Data Mining: A Data Recovery Approach. Chapman and Hall/CRC, 1 edition. (Cited on page 53.)

[Mourelatos and Zhou, 2006] Mourelatos, Z. P. and Zhou, J. 2006. A design optimization method using evidence theory. Journal of Mechanical Design, 128(4):901–908. (Cited on page 57.)

[Murphy, 2000] Murphy, C. K. 2000. Combining belief functions when evidence conflicts. Decision Support Systems, 29(1):1 – 9. (Cited on page 13.)

[NASA, 2011] NASA 2011. Messenger: mission to Mercury. `http://www.nasa.gov/mission_pages/messenger/main/index.html`. Last retrieved 31/12/2011. (Cited on page 120.)

[Neumaier, 2004] Neumaier, A. 2004. Clouds, fuzzy sets, and probability intervals. Reliable Computing, 10(4):249 – 272. (Cited on page 12.)

[Novak and Vasile, 2009] Novak, D. M. and Vasile, M. 2009. A new shaping approach for preliminary low-thrust trajectory design. In: Proceedings of the 21st International Symposium on Spaceflight Dynamics, Toulouse. (Cited on pages xxxii, 144, 145, and 146.)

[Oberkampf and Helton, 2002] Oberkampf, W. and Helton, J. C. 2002. Investigation of evidence theory for engineering applications. In: 4th Non-Deterministic Approaches Forum. AIAA, 1569. (Cited on pages 17 and 25.)

[Orponen, 1990] Orponen, P. 1990. Dempster's rule of combination is p-complete. Artificial Intelligence, 44:245–253. (Cited on page 28.)

[Pareto et al., 1972] Pareto, V., Page, A. N., and Schwier, A. S. 1972. Manual of political economy / by Vilfredo Pareto ; translated by Ann S. Schwier and Alfred N. Page. Macmillan, London. (Cited on page 40.)

[Pate-Cornell, 1996] Pate-Cornell, M. E. 1996. Uncertainties in risk analysis: Six levels of treatment. Reliability Engineering and System Safety, 54:95–111. (Cited on page 25.)

[Phadke, 1995] Phadke, M. S. 1995. Quality Engineering Using Robust Design. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition. (Cited on page 4.)

[Powell, 2001] Powell, M. 2001. Radial basis function methods for interpolation to functions of many variables. In: Lipitakis, E. A. (Editor), Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA-01), Athens, Hellas. HERCMA, LEA Press, Athens, Hellas, pages 2–24. (Cited on page 130.)

[Provan, 1990] Provan, G. M. 1990. A logic-based analysis of Dempster-Shafer theory. International Journal of Approximate Reasoning, 4(56):451 – 495. (Cited on page 28.)

[Rowe and Wright, 2001] Rowe, G. and Wright, G. 2001. Principles of Forecasting: A Handbook for Researchers and Practitioners, chapter Expert opinions in forecasting: the role of the Delphi technique, pages 125–144. Kluwer Academic Publishers. (Cited on page 27.)

[Roy, 2010] Roy, R. K. 2010. A Primer on the Taguchi Method. Society of Manufacturing Engineers. (Cited on page 4.)

[Sacks et al., 1989] Sacks, J., Welch, W., Mitchell, T., and Wynn, H. 1989. Design and analysis of computer experiments. Statistical Science, 4(4):409 – 435. (Cited on pages 55, 130, and 166.)

[Sakata et al., 2004] Sakata, S., Ashida, F., and Zako, M. 2004. An efficient algorithm for kriging approximation and optimization with large-scale sampling data. Computer Methods in Applied Mechanics and Engineering, 193(3-5):385 – 404. (Cited on page 128.)

[Scari, 2005] Scari, E. 2005. Evidence theory applied to the reliable design of interplanetary trajectories. Master's thesis, Politecnico Di Milano. (Cited on page 42.)

[Sentz and Ferson, 2002] Sentz, K. and Ferson, S. 2002. Combination of evidence in Dempster-Shafer theory. In: 6th World Multi-conference on Systemics. Cybernetics and Informatics. (Cited on pages 15, 16, 17, and 32.)

[Shafer, 1976] Shafer, G. 1976. A Mathematical Theory of Evidence. Princeton University Press. (Cited on pages 5, 12, 14, and 16.)

[Smarandache and Dezert, 2005] Smarandache, F. and Dezert, J. 2005. An introduction to the DSm theory for the combination of paradoxical, uncertain

and imprecise sources of information. In: 13th International Congress of Cybernetics and Systems. (Cited on page 12.)

[Spacecraft, 2005] Spacecraft, H. 2005. Image of Itokawa taken on September 10, 2005. http://www.isas.ac.jp/e/snews/2005/0911.shtml. Last retrieved on 28/12/2011. (Cited on page 141.)

[Stanforth et al., 2007] Stanforth, R. W., Kolossov, E., and Mirkin, B. 2007. A measure of domain of applicability for qsar modelling based on intelligent k-means clustering. QSAR and Combinatorial Science, 26(7):837–844. (Cited on page 53.)

[Stone, 1961] Stone, M. 1961. The opinion pool. The Annals of Mathematical Statistics, 32(4):1339–1342. (Cited on page 27.)

[Swiler et al., 2006] Swiler, L., Slepoy, R., and Giunta, A. 2006. Evaluation of sampling methods in constructing response surface approximations. In: Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Newport, RI, United States. 4:2873 – 2896. (Cited on pages 80 and 166.)

[Taguchi, 1978] Taguchi, G. 1978. Off-line and on-line quality control systems. In: Proceedings of the International Conference on Quality Control, Tokyo, Japan. pages 1–5. (Cited on page 3.)

[Taguchi and Phadke, 1984] Taguchi, G. and Phadke, M. 1984. Quality engineering through design optimization. In: IEEE Globecom Conference, New York, USA. Institute of Electrical and Electronics Engineers, pages 1106–1113. (Cited on page 3.)

[Taguchi and Yokoyama, 1993] Taguchi, G. and Yokoyama, Y. 1993. Taguchi methods: design of experiments. TAGUCHI METHODS SERIES. ASI Press. (Cited on page 3.)

[Tessem, 1993] Tessem, B. 1993. Approximations for efficient computation in the Theory of Evidence. Artificial Intelligence, 61(2):315–329. (Cited on page 29.)

[Tholen, 1989] Tholen, D. 1989. Asteroid taxonomic classifications. In: Asteroids II. pages 1139–1150. (Cited on page 142.)

[Trosset, 1997] Trosset, M. 1997. Taguchi and robust optimization. Technical report, Department of Computational & Applied Mathematics, Rice University, Houston, Texas. (Cited on page 4.)

[Vannoorenberghe et al., 1999] Vannoorenberghe, P., Colot, O., and de Brucq, D. 1999. Color image segmentation using Dempster-Shafer's theory. In: International Conference on Image Processing. 4:300–303. (Cited on page 33.)

[Vasile, 2002] Vasile, M. 2002. Robust optimisation of trajectories intercepting dangerous NEO. In: AIAA/AAS Astrodynamics Specialist Conference. AIAA, 4719. (Cited on pages 4 and 35.)

[Vasile, 2004] Vasile, M. 2004. Robustness optimisation of aerocapture trajectory design using a hybrid co-evolutionary approach. In: 18th International Symposium on Space Flight Dynamics. 548. (Cited on pages 33 and 45.)

[Vasile, 2005] Vasile, M. 2005. Robust mission design through evidence theory and multiagent collaborative search. Annals of the New York Academy of Sciences, 1065:152–173. (Cited on pages 35, 42, and 45.)

[Wang, 2003] Wang, G. G. 2003. Adaptive response surface method using inherited latin hypercube design points. Journal of Mechanical Design, Transactions of the ASME, 125(2):210 – 220. Adaptive response surface method;Inherited latin hypercube design points;Central composite design;Computation intensive design problems;. (Cited on page 128.)

[Wang et al., 2005] Wang, G. G., Wang, L., and Shan, S. 2005. Reliability assessment using discriminative sampling and metamodeling. SAE Transactions, Journal of Passenger Cars Mechanical Systems, 114:291 – 300. (Cited on page 128.)

[Yager, 1987] Yager, R. R. 1987. On the Dempster-Shafer framework and new combination rules. Information Sciences, 41:93–137. (Cited on page 16.)

[Yao et al., 2011] Yao, W., Chen, X., Luo, W., van Tooren, M., and Guo, J. 2011. Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. Progress in Aerospace Sciences, In Press, Corrected Proof. (Cited on page 37.)

[Yarnoz et al., 2006] Yarnoz, D. G., Jehn, R., and Croon, M. 2006. Interplanetary navigation along the low-thrust trajectory of BepiColombo. Acta Astronautica, 59(1-5):284 – 293. Space for Inspiration of Humankind, Selected Proceedings of the 56th International Astronautical Federation Congress, Fukuoka, Japan, 17-21 October 2005. (Cited on page 120.)

[Zadeh, 1986] Zadeh, L. A. 1986. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. AI Magazine, 7:85–90. (Cited on page 16.)

[Zadeh, 1999] Zadeh, L. A. 1999. Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 100(Supplement 1):9–34. (Cited on pages 5 and 12.)

[Zhang, 1994] Zhang, L. 1994. Representation, independence, and combination of evidence in the Dempster-Shafer theory, pages 51 – 69. John Wiley & Sons, Inc., New York, NY, USA. (Cited on page 17.)