Design Optimisation of Complex Space Systems Under Epistemic Uncertainty
PhD Thesis

Gianluca Filippi
Aerospace Center of Excellence
Department of Mechanical and Aerospace Engineering
University of Strathclyde, Glasgow

June 5, 2024

Signed: Gianluca Filippi
Date: 23/05/2024

Abstract

This thesis presents an innovative methodology for System Design Optimisation (SDO) through the framework of Model-Based System Engineering (MBSE) that bridges system modelling, Constrained Global Optimisation (CGO), Uncertainty Quantification (UQ), System Dynamics (SD) and other mathematical tools for the design of Complex Engineered and Engineering Systems (CEdgSs) under epistemic uncertainty. The problem under analysis has analogies with what is nowadays studied as Generative Design under Uncertainty. The method is finally applied to the design of Space Systems which are Complex Engineered Systems (CEdSs) composed of multiple interconnected sub-systems. A critical aspect in the design of Space Systems is the uncertainty involved. Much of the uncertainty is epistemic and is here modelled with Dempster Shafer Theory (DST).

Designing space systems is a complex task that involves the coordination of different disciplines and problems. The thesis then proposes a set of building blocks, that is a toolbox of methodologies for the solution of problems which are of interest also if considered independently. It proposes then a holistic framework that couples these building blocks to form a SDO procedure.

With regard to the building blocks, the thesis includes a network-based modelling procedure for CEdSs and a generalisation for CEdgSs where the system and the whole design process are both taken into account. Then, it presents a constraint min-max solver as an algorithmic procedures for the solution of the general Optimisation Under Uncertainty (OUU) problem. An extension of the method for the Multi-Objective Problems (MOP) is also proposed in Appendix as a minor result. A side contribution for the optimisation part refers to the extension of the global optimiser Multi Population Adaptive Inflationary Differential Evolution Algorithm (MP-AIDEA) with the introduction of constraint handling and multiple objective functions. The Constraint Multi-Objective Problem (CMOP) solver is however a preliminary result and it is reported in Appendix.

Furthermore, the thesis proposes a decomposition methodology for the computational reduction of UQ with DST. As a partial contribution, a second approach based on a Binary Tree decomposition is also reported in Appendix.

With regard to the holistic approach, instead, the thesis gives a new definition and proposes a framework for system network robustness and for system network resilience. It finally presents the framework for the optimisation of the whole design process through the use of a multi-layer network model.

0. Abstract

Contents

List of Figures

List of Tables

Acknowledgements

I am grateful to my family for the support given to me in these years.

I would like to thank my supervisor Massimiliano Vasile and all the friends and colleagues I met during this journey: Annalisa Riccardi, Carlos Ortega, Catarina Gomes, Cristian Greco, Gaetano Pascarella, Lorenzo Gentile, Lorenzo Ricciardi, Marilena Di Carlo, Mateusz Polnik, Simao Marto, Tathagata Basu, Victor Rodrigez, Zeno Korondi... too many to list all here.

I would like to acknowledge the support of my funding organisation, the European Commission, through the H2020-MSCA-ITN-2016 Uncertainty Treatment and Optimisation In Aerospace Engineering (UTOPIAE) Marie Curie Innovative Training Network, grant agreement 722734. Thanks to this fellowship, I had the opportunity to attend valuable training events, enlarge my research horizon through formative secondments, and collaborate with promising fellow researchers.

0. Acknowledgements

0. Acknowledgements

AAO All-At-Once
ABM Agent Based Modelling
AIDEA Adaptive Inflationary Differential Evolution Algorithm
AOCS Attitude and Orbit Control System
bpa basic probability assignment
BS Benson Scalarisation
CA Cellular Automaton
CEdS Complex Engineered System
CEdgS Complex Engineered and Engineering System
CEgS Complex Engineering System
CGO Constrained Global Optimisation
CMaOP Constraint Many Objective Problems
CMOP Constraint Multi-Objective Problem
COP Constrained Optimsation Problem
CPSS Chebychev/Pascoletti-Serafini Scalarisation
CS Conic Scalarisation
CSOP Constraint Single-Objective Problem
CSP Constraint Satisfaction Problem
DE Differential Evolution
DES Discrete Event Simulation
DET Direct Energy Transfer
DOD Depth Of Discharge
DS Dynamic Systems
DST Dempster Shafer Theory
DUU Design Under Uncertainty
EA Evolutionary Algorithm
EBRO Evidence-Based Robust Optimisation
EBORe Evidence-Based Optimisation for Resilience
ECS Epsilon-Constraint Scalarisation
EGO Efficient Global Optimisation
ENM Evidence Network Model
FE Focal Element
FOP Free Optimisation Problem

0. Acknowledgements

GEO Geostationary Orbit

GIT Generalised Information Theory

GO Global Optimisation

HCTMC Homogeneous Continuous-Time Markov Chain

IP Imprecise Probability

KPI Key Performance Indicator

LAE Liquid Apogee Engine

LEO Low Earth Orbit

LP Linear Programming

MA Memetic Algorithms

MaOP Many-Objective Problems

MBH Monotonic Basin Hopping

MBSE Model-Based System Engineering

MDD Multi-Disciplinary Design

MDO Multi-Disciplinary Optimisation

MOO Multi-Objective Optimisation

MOP Multi-Objective Problems

MPPT Maximum Power Point Tracker

ML-ENM Multi-layer Evidence Network Model

MP-AIDEA Multi Population Adaptive Inflationary Differential Evolution Algorithm

OBDH On Board Data handling

ODE Ordinary Differential Equation

OUU Optimisation Under Uncertainty

OR Operational Research

PSS Pascoletti-Serafini Scalarisation

PT Probability Theory

PCDU Power Conditioning and Distribution Unit

QoI Quantity of Interest

RBDO Reliability Based Design Optimisation

RDO Robust Design Optimisation

ReBDO Resilience Based Design Optimisation

RFDN Radio Frequency Distribution Network

SD System Dynamics

SE Systems Engineering

0. Acknowledgements

SDO System Design Optimisation
SOP Single-Objective Problem
SoS System of Systems
SR Success Rate
UBD Uncertainty-Based Design
UMDO Uncertainty-Based Multi-Disciplinary Design Optimisation
UQ Uncertainty Quantification
TTC Telemetry, Tracking and Command System
WCS Weighted Chebichev Scalarisation
WSS Weighted-Sum Scalarisation

0. Acknowledgements

0. Acknowledgements

1

Introduction

" Begin at the beginning, the King said, gravely,
and go on till you come to an end;
then stop. "

– Lewis Carroll, *Alice in Wonderland*

There is a tendency in the evolution of science and in general in our artificial world at complexification through alternated phases of specialisation and revolution [1], the latter as in the definition of Kuhn. This process has many analogies with what happens in Nature. The last paradigm shift in science and technology, up to now, is happening in these decades, but some thinkers were able to foresee it. For example, in his paper "Science and Complexity" [2] in 1948 Warren Weaver gave a prediction of the type of problems the next generation of scientists and engineers would have dealt with. After solving "Problems of Simplicity" before 1900 and, in the opposite direction, "Problems of Disorganised Complexity" in the first half of the 20th century, the challenge for the coming years, he thought, would have been represented by problems of "Organised Complexity". Also Stephen Hawking correctly said during an interview in 2000: "the next century will be the century of complexity". Their intuition was correct and we can acknowledge this also in what is happening nowadays within the field of Systems Engineering (SE). Indeed, recent advances in science and technology have led to a rapid increase in the complexity of most engineering problems and in many cases this change has been a qualitative one rather than merely one of magnitude. As stated by Alessandro Vespignani in "L'algoritmo e l'oracolo" [3], two major scientific revolutions (or paradigm's shifts using Kuhn's words) happened to shape our modern world: the rise of "Complexity" and "Complex Science" and the "Datification", the accumulation and exploitation of huge amount of data as it is continuously produced by sensors, social platforms, etc.

Therefore, the challenge for today and the coming years is to develop a new mindset and a new set of engineering tools and methodologies that will be able to deal and govern this increasing complexity [4, 5]. We are however still in the infancy of this transition since it is difficult even to give a single and accepted definition of "Complexity" [6–8].

Engineers have to face, control and further design many different types of systems that have been accumulating in the centuries: Simple Systems, System of Systems (SoS), CEdS, Complex Engineering System (CEgS) and CEdgS. The last three

types of these systems are the ones causing the challenge and they are also the interest of the thesis. Simple systems can be understood through the lens of 'Problem of Simplicity' of Warren Weaver [2]. They can be modelled with few parameters and they accept to be designed and controlled by traditional methodologies. SoS still can be re-conducted to this category and are systems which parts can be further decomposed, at many levels, in other (sub)systems. They are complicated and usually made by a big number of components. However, their global behaviour can be understood as the sum of the system's parts. The reductionist approach based on the system decomposition and parallel analysis of the components is suitable for them. Other engineered systems can be included in the class of 'Disorganised Complexity' [2] that can be thought as the opposite of 'Problem of Simplicity'. In Weaver's view indeed, disorganised complexity results from the particular system having a very large number of parts, say millions of parts, or many more. Though the interactions of the parts in a "disorganised complexity" situation can be seen as largely random, the properties of the system as a whole can be understood by using probability and statistical methods. A prime example of disorganised complexity is a gas in a container, with the gas molecules as the parts. Finally, there are engineered systems that can be classified as 'Organised Complexity' [2]. Organised complexity is primarily found in the interconnected and non-random interactions among the components. These interrelated connections give rise to a coordination due to hierarchies and structure and the system exhibits characteristics that are not inherent in or controlled by its individual components. These systems are made by a number of heterogeneous sub-systems that interact non-linearly at different levels of organisations producing an emergent global behaviour that can not be predicted by a reductionist approach. A characteristic of this new class of problems is their evolvability in the sense that the CEdS is able to adapt after internal and/or external variations by auto-organising itself. It is worth noting that a system doesn't necessarily require a large number of components to exhibit emergent properties. These systems are usually defined CEdS, CEgS and CEdgS. As Sheard et al. describes in [9] they can be seen at the border of the order-to-chaos spectrum.

The increase of number and types of engineered systems has proceeds at an equal pace with the development of SE design methodologies. Chapter 2 will present the evolution of engineering methodologies in the last centuries to cope with an increasing body of knowledge and available tools. It will explain the progressive automation of the design process by the inclusion of analysis tools [10], multi-disciplinari methods as SDO and Multi-Disciplinary Optimisation (MDO) [11, 12], UQ for Design Under Uncertainty (DUU) and OUU [13, 14], optimisation frameworks [15] and specialised optimisation approaches based on metrics that have been studied specifically for CEdgS as "Robustness" in Robust Design Optimisation (RDO) [11, 16–21] and "Reliability" in Reliability Based Design Optimisation (RBDO) [12].

The same chapter describes however how the current technologies have reached the limit and a qualitative shift is needed to cope with complexity. Extreme, rare and unpredictable events are unfortunately becoming more frequent in networked systems and as explained by Crucitti et al. in 'Model for cascading failures in complex networks' [22] cascades of failures can be triggered in CEdS by small initial shocks when systems are composed of interconnected sub-systems or inter-dependent networks.

To overcome this gap, there is a sound agreement that different tools from scientific fields have to be integrated. In [23] an analogy with biological systems is used to define CEdS as 'living' and adapting systems that can change and react to inputs from the environment and also from internal unexpected events. The authors suggest an overarching graph theory approach on top of the engineering modelling and optimisation process to investigate the resilience of CEdS. "Resilience" is then a promising metric which quantification could permit to better understand the relation between structure and dynamics of CEdgSs. Attempts to define 'Resilience' can be found in 'Framework for analytical quantification of disaster resilience' by G. P. Cimellaro et al. [24], 'A review of definitions and measures of system resilience' by S. Hosseini et al. [25], 'Essential characteristics of resilience' by D. Woods et al. [26] and 'Essentials of resilience, revisited' by [27], D. Woods et al. A method for incorporating "Resilience" in the design framework of CEdgS is then Resilience Based Design Optimisation (ReBDO) [13,23,28].

This thesis then proposes a novel methodology to cope with this problem. In agreement with the research community results as discussed above, it presents a holistic method for SDO with the final goal to do OUU and to design CEdgSs for Resilience. In particular it is related to MBSE, that is a SE [29] methodology that uses, for better analysis and design, models to represent the system's components, behaviour, and interconnections [30]. The design process is applied to both the system itself and the engineering design process. A space mission is indeed developed progressing through a series of macro-phases: concept exploration, detailed development, production and deployment and operations and support [31] and this is a long process that can take up to 10 or more years. While the process proceeds, knowledge increase, uncertainty is reduced but also external and stake-holder conditions can change.

With this goal in mind, different tools and theories have been used and combined in this thesis in order to build an holistic framework for SDO. The most important are: Global Optimisation (GO) and in particular CGO, UQ with the use of DST and network modelling approaches. The general goal is to use DST to capture and properly model epistemic uncertainty, to use the network modelling approach through ENM and Multi-layer Evidence Network Model (ML-ENM) to obtain a fast estimation of the impact of uncertainty in the system model and finally to merge these components together in an automated OUU framework to decide by design the optimal system configuration under lack of knowledge.

GO [32–35] gives the general structure for the automation of the design process. Meta-heuristics algorithms, in particular, will be used [36]. Requirements have also to be included in the optimisation process and this is done with the handling of constraints [36–39]. The thesis will explain how the engineering problem can be translated to a corresponding optimisation one. It will focus on CSOP but some results in the Appendix are reported also for CMOP. The CGO method is coupled with a network system model to perform SDO [40] dealing with different coupled disciplines through an holistic and comprehensive point of view. CGO is also coupled with UQ to understand how uncertainty can affect the final design of the system, how this effect is correlated with the choice of the design parameters and how to select the optimal system configuration in the worst condition given by uncontrolled factors. This coupling, with suitable metric definition as in the following chapters, brings to the RDO prob-

lem which will be solved through the worst-case optimisation formulation [41–43] and to the ReBDO for which two alternative approaches will be explained, one based on Catastrophe Theory [44, 45] and the other on Stochastic Processing.

UQ [46–48] is then an important component of this thesis. Uncertainty comes in different forms and from different sources. There is uncertainty in the models' parameters and variables, in the environment and in the human interventions. Two macro-categories of uncertainties can be identified: aleatory uncertainty and epistemic uncertainty [49]. The former is due to natural randomness which cannot be reduced while the latter is due to the lack of information/knowledge or incomplete data. Epistemic uncertainty is reducible by acquiring more knowledge on the problem. Both these categories have to be considered during the design process of a space system. Epistemic uncertainty is particularly important in the early design phases. Knowledge about systems and requirements is indeed only acquired incrementally during the design process, but substantial commitments are made upfront, essentially in the unknown. DST [50, 51], within the set of Imprecise Probability (IP) theories, will be used and its coupling with the optimisation framework will permit to perform OUU.

A network modelling approach [52, 53] is also used to do SDO, as noticed above. It is based on what has been defined ENM and ML-ENM. The network modelling is important because it drastically reduces the computational complexity of UQ with DST.

The methodology is finally applied to the design of Space Systems. They are complex systems since they include multiple interconnected components and disciplines with non linear couplings: payloads, structure, thermal system, attitude, control, etc.

The remaining of this section presents the structure of this introductory chapter. Section 1.1 explains the motivation and rationale of the thesis by presenting the limitations of current engineering methodologies. Section 1.2 presents then the research objectives and contributions of this thesis. Section 1.3 gives the list of academic publications. Finally Section 1.4 summarises the structure of the following chapters.

1.1 Motivation and rationale

Even if the research in fields such as GO, CGO, OUU, IP and SDO is very active, industries usually have a conservative approach based on traditional methods like margins and redundancies [54]. Margins are used to break the endless loop that would be generated trying to optimise the space system at different granularity definitions, as knowledge increases. However, these conservative approaches introduce limitations on the design process as explained in the previous section. In the current age of increasing complexity, indeed, the margin approach lacks the capability to model interconnectedness, to understand how uncertainty propagates and how dynamics arise from the structure of a CEdgS.

The limitations we aim to solve with the proposed methodologies in this thesis are:

1. **Improper treatment of uncertainty**. The lack of an appropriate quantification of uncertainty (with the use of margins and contingencies or classical probability theory) brings to an overestimation or an underestimation of the effect of

uncertainty. This increases costs and development time or the occurrence probability of undesirable events.

2. **High computational cost of UQ**. There are many reasons why proper quantification of uncertainty, in particular within the framework of IP, is not implemented by industries. One of the most critical is related to its computational costs. Cost-efficient tools for UQ are indeed missing.

3. **Lack of a holistic view**. The reductionist approach is still dominant. Some approaches for system-level optimisation including uncertainty exist, however, they still present some critical limitations.

1.2 Research Objectives and Contribution

The general goal of this thesis is to help overcome the limitations described in the section above. The thesis' objectives are summarised in the following list.

1. Development of a modelling framework for capturing the complexity of the CEdgS. This includes:
   - a network-based modelling framework of the CEdS that should model the dependencies between sub-systems and the system uncertainty. The model should allow us to do UQ with DST;
   - a multi-layer network-based modelling framework of the CEdgS that by extending the one at the previous point, should model both the system and its evolution during the design process.
2. Development of algorithmic methods for UQ within the framework of DST for the quantification of epistemic uncertainty:
   - the methods should use DST to quantify epistemic uncertainty in the CEdS;
   - the methods should quantify the propagation of epistemic uncertainty through the different phases of the design process;
   - the methods should exploit the network structure of the model to reduce the computational cost of UQ with DST.
3. Development of a novel definition of 'Resilience Engineering' and of new metrics of 'Resilience'.
4. Development of algorithmic procedures for the solution of CGO problems:
   - the thesis should develop or extend methods for single-layer global constraint optimisation;
   - the thesis should develop or extend methods for global constraint optimisation under uncertainty;
5. Development of a holistic approach for SDO which combines all the building blocks listed above.

- the thesis should develop a SDO framework for RDO under epistemic uncertainty to optimise the 'Robustness' of the system.
- the thesis should develop a SDO framework for resilient optimisation under epistemic uncertainty.
- the thesis should develop a SDO framework for the design optimisation of the whole process. It should take into account the system and its evolution through the design process.

1.3 Publications

Journal Publications

1. **G. Filippi**, M. Vasile, D. Krpelik, P. Z. Korondi, M. Marchi, and C. Poloni, Space systems resilience optimisation under epistemic uncertainty, *Acta Astronautica*, vol. 165, pp. 195210, 12 2019, https://doi.org/10.1016/j.actaastro.2019.08.024 [13].

2. **G. Filippi** and M. Vasile, Global Solution of Constrained Min-Max Problems with Inflationary Differential Evolution, in *Optimisation in Space Engineering OSE (E. Minisci, A. Riccardi, and M. Vasile, eds.), no. Optimization and Engineering, Springer*, 2020, https://doi.org/10.1007/s11081-021-09613-3 [55].

Book Chapters

- **G. Filippi** and M. Vasile, A multi layer evidence network model for the design process of space systems under epistemic uncertainty, in *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences (A. Gaspar-Cunha, J. Periaux, K. C. Giannakoglou, N. R. Gauger, D. Quagliarella, and D. Greiner, eds.)*, (Cham), pp. 227243, Springer International Publishing, 2021 *https://doi.org/*10.1007/978−3−030−57422−2_15 [56].

- **G. Filippi** and M. Vasile, Network resilience optimisation of complex systems in *Advances in uncertainty quantification and optimization under uncertainty with aerospace applications (M. Vasile and D. Quagliarella, eds.)*, (Cham), Springer International Publishing, 2021, *https://doi.org/*10.1007/978−3−030−80542−5 [57].

- **G. Filippi** and M. Vasile, Introduction to Evidence-Based Robust Optimisation, in *Optimization Under Uncertainty with Applications to Aerospace Engineering (M. Vasile, ed.), Springer Nature*, 2021, *https://doi.org/*10.1007/978−3−030−60166 − 9_17 [58].

- **G. Filippi** and M. Vasile, Inflationary Differential Evolution for Constrained Multi-Objective Optimisation Problems, in *Bioinspired Optimisation Methods and Their Application*, pp. 2942, Springer, Cham, 11 2020, *http://link.springer.com/*10.1007/978 − 3 − 030 − 63710 − 1_3 [59].

Peer-Reviewed Conference Papers and Presentations

- C. O. Absil, M. Vasile, **G. Filippi**, A. Riccardi, and M. Vasile, A Variance-Based Estimation of the Resilience Indices in the Preliminary Design Optimisation of Engineering Systems Under Epistemic Uncertainty, in *EUROGEN, (Madrid)*, 2017, https://www.researchgate.net/publication/323242901 [60].
- M. Vasile, **G. Filippi**, C. Ortega Absil, and A. Riccardi, Fast belief estimation in evidence network models, in *EUROGEN, (Madrid)*, 2017, https://pureportal.strath.ac.uk/en/publications/fast-belief-estimation-in-evidence-network-models [61].
- **G. Filippi**, M. Marchi, M. Vasile, and P. Vercesi, Evidence-Based Robust Optimisation of Space Systems with Evidence Network Models, in *IEEE Congress on Evolutionary Computation (CEC), (Rio De Janeiro)*, 2018 [62].
- **G. Filippi**, M. Vasile, P. Z. Korondi, M. Marchi, and C. Poloni, Robust design optimisation of dynamical space systems, in *8th International Systems & Concurrent Engineering for Space Applications Conference*, (Glasgow), 2018 [63].
- **G. Filippi**, D. Krpelik, P. Z. Korondi, M. Vasile, M. Marchi, and C. Poloni, Space systems resilience engineering and global system reliability optimisation under imprecision and epistemic uncertainty, in *Proceedings of the International Astronautical Congress, IAC, vol. 2018-Octob, (Bremen)*, 2018 [64].
- **G. Filippi** and M. Vasile, Evidence-based resilience engineering of dynamic space systems, in *Proceedings of the International Astronautical Congress, IAC*, vol. 2019-Octob, (Washington), 2019 [65].
- **G. Filippi** and M. Vasile, "A multi layer evidence network model for the design process of Space Systems under epistemic uncertainty", EUROGEN, 2019 [66].
- **G. Filippi** and M. Vasile, A Memetic Approach to the Solution of Constrained Min-Max Problems, in *2019 IEEE Congress on Evolutionary Computation, CEC 2019* - Proceedings, (Wellington), pp. 506513, 2019 [67].
- C. Greco, L. Gentile, **G. Filippi**, E. Minisci, M. Vasile, and T. Bartz-Beielstein, Autonomous Generation of Observation Schedules for Tracking Satellites with Structured-Chromosome GA Optimisation in *2019 IEEE Congress on Evolutionary Computation, CEC* 2019 - Proceedings, (Wellington), pp. 497505, 2019, https://doi.org/10.1109/CEC.2019.8790101 [68].
- **G. Filippi**, D. Gillespie, A. Ross Wilson, and M. Vasile, A resilience approach to the design of future Moon base power systems, in *Int. Astronaut. Congr. IAC*, 2020 [69].
- **G. Filippi**, M. Vasile, E. Patelli and M. Fossati, "Generative Optimisation of Resilient Drone Logistic Networks", WCCI, 2022 [70].
- **G. Filippi**, M. Vasile, E. Patelli and M. Fossati, "Multi-layer Resilience Optimisation for Next Generation Drone Logistic Networks", ESREL, 2022 [71].

Conference Papers and Presentations

- D. Gillespie, A. R. Wilson, D. Martin, G. Mitchell, **G. Filippi**, M. Vasile, 'Comparative analysis of solar power satellite systems to support a moon base', IAC 2020 [72].
- L. Gentile, **G. Filippi**, E. Minisci, M. Vasile, T. Bartz-Beielstein, and M. Vasile, Preliminary spacecraft design by means of Structured-Chromosome Genetic Algorithms, in IEEE Congress on Evolutionary Computation (CEC), Glasgow, 2020 [73].

Prizes

- First running up, World Congress on Computational Intelligence (WCCI), 2019
- Complex systems and smart cities paper award, European Conference On Safety And Reliability (ESREL), 2022.

1.4 Thesis Structure

After this introductory Chapter 1, the thesis is divided into three main blocks, where Parts II and III represent the main contribution. Conclusions are then in Chapter 10. Three Appendix chapters are given for completeness at the end: Appendices A to C.

Part I presents an overview of the required theoretic background:

- (Chapter 2): *Present and Future of System Engineering.* This chapter gives an introduction to System Engineering and to the history of design approaches. It explains why in our age of complexity a qualitative shift is required about tools and methodologies used by designers. It presents the current research direction and focuses finally on the need of holistic approaches and on the concept of Resilience Engineering.
- (Chapter 3): *Evidence-based Uncertainty Quantification.* This chapter gives an introduction to UQ. It presents different problem formulations, the most important theories and then the algorithmic approaches in the literature. The focus of the chapter is on IP and in particular in DST.
- (Chapter 4): *Optimisation Under Uncertainty.* The chapter gives an introduction to Optimisation. It presents the most important problem formulation for the topic considered in this thesis. It presents the algorithmic approaches in the literature. The most important points of this chapter are: constraint global optimisation, optimisation under uncertainty, worst-case optimisation, multi-disciplinary optimisation and constraint handling.

Part II presents novel algorithmic approaches for the solution of constraint global optimisation problems. The part includes a single chapter:

- (Chapter 5): *Global Solution of Constrained Min-Max Optimisation.* It is here presented the constraint single-objective worst-case scenario optimisation (min-max) procedure. The methodology is coupled with a generalised version of the global optimisation solver MP-AIDEA.

Part III proposes finally the developed holistic methods for SDO under epistemic uncertainty with the use of DST:

- (Chapter 6): *Evidence Network Model for System Design Optimisation.* It is here presented the ENM, a network model for CEdS that captures epistemic uncertainty using DST.
- (Chapter 7): *Evidence-based Robust Optimisation.* It is here presented the Evidence-Based Robust Optimisation (EBRO) approach for the optimisation for Robustness of CEdS. The method includes also a network-decomposition approach that is based on the ENM for a fast quantification of Belief and Plausibility within DST. The method can be thought of in analogy to decomposition techniques for MDO.
- (Chapter 8): *Evidence-based Resilience Optimisation.* It is here presented Evidence-Based Optimisation for Resilience (EBORe) for the resilience optimisation of CEdS under epistemic uncertainty. This part introduces also a new definition of Resilience Engineering and two alternative modelling approaches of system resilience, based respectively on the Stochastic Processes Markov Chain and on the Catastrophe Theory.
- (Chapter 9): *Multi-layer Network Model for Design Process Optimisation Under Epistemic Uncertainty.* This chapter introduces ML-ENM, a generalisation of ENM, and an approach for the optimisation of the CEdgS through the whole design process. Each layer of the ML-ENM represents a different phase in the design process, each node represents a subsystem or a component at a particular level of granularity and each link is a sharing of information.

Other contributions to the thesis have been finally added in the Appendix:

- (Appendix A): *Testing Results for the single-objective min-max problem.* Includes more detailed test results for the min-max algorithm.
- (Appendix B): *Multi objective min-max approach* presents a generalisation of the min-max approach to the CMOP.
- (Appendix C): *Outer Belief Estimation via Evolutionary Binary Tree.* describes an alternative decomposition approach for UQ with DST that makes use of Binary Tree Decomposition

It has been decided to present this last material in a separate Appendix section because, in order, (A) is merely a long list of numbers and tables, (B) needs more refinement to be completed and (C) is not the result of my only work.

Part I

Background

2

Present and Future of Systems Engineering

" If you fail to plan, you plan to fail "

– Bejamin Franklin

SE is a trans-disciplinary and integrative approach to enable the successful realisation, use, and retirement of engineered and engineering systems. It uses systems principles and concepts, and scientific, technological, and management methods. Not only it integrates different disciplines and analysis, but also it 'transcends' them to define a higher level structure and organisation, both technical and economical, used to drive the design and production of the engineered system 'from the cradle to the grave'. Indeed, systems engineers model and control the early concept definitions, the design phase, the system's operation and finally the system disposal. The SE approach is particularly useful for identifying, at the early stages, future risks about cost, time of the process and possible other Quantity of Interests (QoIs) [29]. This problem-solving attitude has been applied, with an increasing level of efficiency and consciousness, since the first human civilisations. Knowledge and expertise have been accumulated over the centuries, making solving more complicated problems possible and being 'cheaper and faster'. However, we have achieved in the last years the saturation of this process and a fundamental qualitative shift in the engineering approach is now required [4, 9, 23]. Our current methodologies are indeed crashing against the increasing complexity of our human world. Most of the problems SE is now facing are not composed anymore of SoS for which a reductionist approach is feasible, but instead of CEdS, CEgS and in general of CEdgS which require a holistic way of thinking.

This chapter presents the necessary background to understand SE. It focuses in particular on MBSE [29] which makes use of models to represent the system's components, behaviour, and interconnections. It presents the current state of the art, the history and the critical points of MBSE. It explains the current challenges to evolve new methodologies to deal with an always increasing level of complexity in CEdgS. It states then the need for an integration of optimisation, UQ and other mathematical tools. It introduces Resilience Engineering which is vastly recognised as the most promising direction for SE.

This chapter is a necessary introduction to the next background Chapters 3 and 4 in Part I where the aspects of UQ and optimisation are expanded respectively. The chapter is also linked to the thesis' contributions in Parts II and III and in particular to the modelling framework in Chapters 6 and 9 and to the system design optimisation approaches EBRO in Chapter 7 and EBORe in Chapter 8.

The chapter is organised as follows. Section 2.1 gives an overview of the historical evolution of the design approaches for engineering systems. Section 2.2 gives a description of the project life cycle for the design of space systems. Section 2.3 lists a few examples of disasters in CEdS showing that the traditional approaches to MBSE are not effective anymore with the increase of complexity in the last years. Section 2.4 presents the future trend in the development of system design approaches and introduces Resilience Optimisation. Section 2.5 introduces some of the mathematical tools used in the following of the thesis. Section 2.5 focus finally on models by presenting modelling methodologies for engineered (but not only) systems, model typologies, modelling and simulation paradigms and modelling languages.

2.1 History of the Systems Design

Engineering design approaches have been evolved in time at an equal pace with the development of technology and particularly with computational power. This section lists and briefly describes the fundamental steps in this evolution. The following methods can be viewed as an historical progression towards the most general, efficient and powerful approach for SE. However, since no perfect methodology exists for solving all possible problems and since engineers continuously face situations with a highly variegated characteristics and difficulties, this list can also be understood also as the accumulation of a set of tools that can be applied each one in a different situation.

We can define the first approach as 'Design by Formula' which is based only on the calculations made by the designer and possibly on the feedback given by a physical prototype. Software support is not used. Numerical methods are instead introduced in the 'Design by Analysis' approach which shorten the design process and enable a better understanding of the problem without, or reducing, the use of expensive physical experiments. The design process is still based on a reductionist approach where subsystems and components are considered separately without particular attention to their interfaces and reciprocal effects. With 'Design by Optimisation' [15], instead, a system-level design is handled with the use of numerical optimisation algorithms integrated with analysis tools. This allows the designers to automatise some of the procedures and to limit the human-in-the-loop. In such an integrated approach it is important to define models of the engineered system which can be coupled with the optimisation frameworks. This has brought to MBSE and to the algorithmic framework of Model Based Optimisation (MBO) [74, 75]. The model highlights the specific characteristics of interest and then is used as black or grey box in order to find the optimal parameter configuration for given objective and constraint functions.

A few interdependent directions have been then explored. On one hand, there is indeed interest in dealing with the coupling of different fields or disciplines. This have brought to Multi-Disciplinary Design (MDD) and more specifically to mathemat-

ical/optimisation tools such as MDO and SDO. In this respect, it has to be highlighted that there is a fundamental difference between the two [11,12]. In MDO a small number of different disciplines are strongly coupled while in SDO there could be many different fields which are however partially or weakly coupled. This thesis, in particular, is focused on SDO. On the other hand, since CEdgSs are inevitably affected by uncertainty and imprecision, it is important to understand and capture its effect on the system. This has brought to the Uncertainty-Based Design (UBD) approach [76, 77] and in particular to optimisation techniques under the definition of OUU [14].

With regards to UBD, various Key Performance Indicators (KPIs) have been considered and studied. One of these is 'Robustness' which can be defined as the property of the system for which its structure (or topology from a network point of view) is more important than dynamics. Optimisation techniques called RDO [11,16–21] have indeed been developed in order to design the structure of the system such that it minimise the risk related to uncertainty to reduce the system's performance. An other important metric is 'Reliability' which is the probability that a system will perform properly for a specified period of time under a given set of of operating conditions [78]. Optimisation approaches have been developed for this, called RBDO [12]. 'Resilience' is finally a new concept that start to be investigated. It is considered as the capability of the system to recover from expected or unexpected disturbances, bouncing back not necessarily exactly to the previous performance. Here dynamics over the system (network) is more important than its structure (topology). The interest in this concept has brought to the design approaches of ReBDO [13, 23, 28]. The ReBDO is presented in more detail in Section 2.4 and the this contribution about resilience optimisation, called EBORe, is in Chapter 8.

Moreover, a further important step is given by the application of the coupling between the holistic MDD and UBD giving birth to Uncertainty-Based Multi-Disciplinary Design Optimisation (UMDO) theory [79].

Despite the important progress in the methodologies for system design, it is however common practise in the Space industry to apply the well-established safety margins and redundancies approach. The margin approach evaluates the quantity of interest (for example the mass) associated with a proposed nominal design solution, called Best Estimate (BE), and adds to it, and to each subsystem quantity of interest, a margin often called the *contingency* or *safety factor*. The safety factor accounts for the expected variations of all uncertain components. For example, a margin is added to the power demand when sizing the solar arrays on a spacecraft. The value of the quantity of interest at the system level, after margins are applied, is often called the Maximum Expected Value (BE). The difference between the Maximum Expected Value (BE) and the Maximum Possible value (MP) is generally considered to be a further margin that accounts for the unexpected variation of all uncertain components [54]. This traditional method, however, lacks an appropriate quantification of uncertainty. As a consequence, there can be an overestimation or an underestimation of the effect of uncertainty which can lead to either an increase in costs and development time or to the occurrence of undesirable events. As it was recognised during the Columbia Accident Investigation Board (CAIB) [80], the classic pattern that brings to failure, common to many other tragic accidents [81], is the combination of production pressure, that

pushes to reduce the safety margins, and a fragmented problem solving that lacks a system-level understanding. Hence one can argue that there is the need to incorporate a proper quantification of uncertainty and of system complexity within the systems engineering approach [74, 82].

2.2 Space Systems Project Life Cycle

The design of a space mission is a complex and time consuming process. Indeed, it can requires up to several years from early conception to launch and commissioning. Also, it involves different players with different goals: end users or customers, operators, scientists, developers and sponsors. In order to decompose the whole design process into smaller and more manageable pieces, the life cycle of a space mission traditionally proceeds through four main phases [31]:

1. Concept exploration phase: broadly defines the space mission and its components.
2. Detailed development phase: defines more precisely system's components.
3. Production and deployment phase: constructs and launches the system.
4. Operations and support phase: daily supports the mission and brings it to its end of life.

Depending on the mission's sponsor - National Aeronautics and Space Administration (NASA), European Space Agency (ESA), commercial enterprise - these phases are further divided and labelled differently. Phases are separated by Key Decision Points (KDPs), milestones where the authority, based on the progress state, the achieved results, the requirements and the budget, approves or rejects the project with a "go" or "no go" decision. For example, NASA in the 'Systems Engineering Handbook' divides the project life cycle into the following steps [83, 84]:

1. Pre-Phase A. Initial studies and concept development are conducted to determine the feasibility of a project. This includes identifying potential missions and defining high-level objectives, as well as assessing technical, schedule, and cost risks.
2. Phase A. The mission concept is developed, including mission objectives, requirements, and constraints. The system architecture is defined, and initial analyses and trade studies are conducted to support decision-making.
3. Phase B. The focus is on the preliminary design of the mission and its subsystems. A detailed mission plan is developed, and requirements and specifications are refined. The design is evaluated against technical, cost, and schedule constraints, and risks are identified and managed.
4. Phase C. The detailed design of the mission and its subsystems is completed, and hardware and software components are developed and tested. The mission plan is finalised, and the system is integrated and tested to verify that it meets all requirements.

5. Phase D. The mission is launched and operated. Mission data is collected and analysed, and any anomalies or issues that arise are addressed. The mission is also evaluated against its original objectives to assess its success.

6. Phase E. Mission is decommissioned. This includes the disposal of any hardware or software components and the closure of any facilities or resources associated with the mission.

A similar approach is used by ESA that in the the 'Project Phasing and Planning' document from the European Cooperation for Space Standardisation (ECSS) [85] defines the following phases:

1. Phase 0: Mission Analysis/Needs Identification. This is a preliminary phase where the different stakeholders agree on a first definition of the mission. Needs, expected performance, operating constraints and safety goals are defined. Are here identified possible system concepts. It is produced the assessment of project management data (organisation, costs, schedules). At the end of the phase 0, a Mission Definition Review (MDR) can take place.

2. Phase A: Feasibility. The feasibility phase should result in finalising the expression of needs expressed in phase 0 and proposing solutions meeting the perceived needs. During this phase, technical solutions for the system concept selected in phase A are decided.

3. Phase B: Preliminary Definition (Project and Product). From a system level point of view, the System Requirements Review (SRS) is conducted. In this phase analysis are performed to accumulate information. First decisions on Make or buy are done. There is also a confirmation of project feasibility.

4. Phase C: Detailed Definition (Product). There is the first production of representative elements of the selected solution, leading to a detailed definition of the system and its components. At the end of this phase, the Critical Design Review (CDR) is conducted.

5. Phase D: Production/Ground Qualification Testing. This is the final stage of the system development. Ground qualification testing are performed. Usually phases C and D are combined. This phase ends with the Acceptance Review (AR).

6. Phase E: Utilisation. This phase comprises the launch campaign, launch and in-flight acceptance of space elements if needed, and it corresponds to operation and maintenance of the system, as well as the acquisition of feedback.

7. Phase F: Disposal. The disposal phase covers all events from the End-of-Life (EOL) till final disposal of the product.

2.3 Complexity and Limits of Current Approaches

As foreseen by Warren Weaver in 'Science and Complexity' [2] we entered in the phase he defined of 'Organised Complexity' where systems are qualitatively different from

the ones Systems Engineers used to design and 'complexity' is the key point. What is complexity? This section tries first to give a definition. Then it provides a non exhaustive list of disasters in CEdS that are driven by it. The section should make clear the need of novel methodologies for the design of CEdgS which could overcome the problem of complexity by a qualitative shift in the engineering approaches.

It is difficult to converge to a single definition of 'Complexity' since many of the experts working in the filed have their own definition that often don't even match. Seth Lloyd came up with more than forty alternative definitions [6]. Melanie Mitchell in her book 'Complexity: A guided tour.' [7] describes one of the first meetings she organised on complexity. As first question to the attendees, she asked to give a definition of the concept. The session ended up with a long list of different and incompatible definitions. Some characteristics can be however determined to be shared by complex systems of various nature and accepted by experts in different fields. Indeed, David Snowden and Mary Boone [8] describe complex systems as having the following characteristics: (i) Large number of interacting elements; (ii) Non-linear interactions such that seemingly minor changes can produce significant consequences; (iii) Solutions emerge from dynamic circumstances, that is, they are emergent; (iv) Elements of the system evolve together in irreversible ways; (v) Hindsight cannot lead to foresight because conditions and the system constantly change; (vi) The system and the agents operating within it constantly operate on each other in ways that are unpredictable. It has to be noticed that 'large number' in this context has to be understood as a quantity between the small set of 'Problem of Simplicity' and the huge set of 'Problem of Unorganised Complexity' as explained in [2].

Since complexity increases in CEdgSs, extreme, rare and unpredictable events are unfortunately becoming more frequent. These events are usually called 'Black Swans' or 'Dragon Kings'. The former refer to events that are rare, unexpected, and have a significant impact. The term was popularised by Nassim Nicholas Taleb in his book 'The Black Swan' [86] and is based on the idea that, before the discovery of black swans in Australia, it was believed that all swans were white. 'Dragon Kings' are even more extreme events with greater impact than Black Swans. The term was coined by Didier Sornette and colleagues in their research on financial crashes and other extreme events.

In the following, some examples of Black Swans that happened to network systems are described where the disasters were triggered by the failure of a single node or link:

- October 1986: during the first documented Internet congestion collapse, the data throughput from LBL to UC Berkeley (separated by less than 400 meters and three Interface Message Processor (IMP) hops that correspond to the modern routers) dropped from 32Kbps to 40bps [87–89].

- August 1996: 1300-megawatt electrical line in southern Oregon sagged in the summer heat. In an electrical grid, when for any reason a line goes down, its power is automatically shifted to the neighbouring lines, which in most of the cases are able to handle the extra load. Sometimes, however, these lines are also overloaded and must redistribute their increased load to their neighbours. In this case the failure propagated through the network eventually causing the black-out in 11 Western States [22].

- 14 August 2003: initial disturbance in the electrical grid in Ohio triggered a large area black-out. People remained without electricity for as long as 15 h [22, 90].
- February 2010: the snowstorms along the Eastern coast of the US caused the cancellation of more than 20,000 ights (4.2 % of all flights) scheduled during the month with the peak of 23 % on the 10 February. This large number was mainly due to the near or complete closure of on a few Northeastern hub airports. The total cost of these cancellations is unknown due to the difficulties in quantifying the exact cost to each carrier; however, across all carriers it has been estimated that cancellations from snowstorms over the whole month cost between $ 80100 million [91].
- 18 December 2010: Heathrow airport was closed for arrivals and departures with only a limited number of flights operating the next day, due to 70 mm of snow falling in one hour. This event caused the cancellation of over 4000 flights, disrupting the travel plans of many passengers during what was predicted to be Heathrows busiest weekend of the year [91].

More complex networked systems can be modelled as Inter-dependent and heterogeneous systems of networks [92]. The following is a list of Black Swans that happened to these multi-layer network systems:

- 28 September 2003: an important blackout in Italy was triggered by the bidirectional coupling between Information Communication Technology (ICT) and energy networks [93, 94].
- 2007: the UK floods led to the inundation of energy and water facilities in the flood plain. This subsequently led to a regional loss of these services as well as the loss of electricity-dependent Information Communication Technology (ICT) networks and reduced emergency response capacity as a result of transport network disruption [93].

2.4 Complexity and Resilience Engineering

This section introduces to the connection between complexity and CEdgS. Then, it presents the most promising research direction for controlling and driving complexity through the engineering design process. This refers to the concepts of 'Resilience' and 'Resilience Engineering'.

As explained by Crucitti et al. in 'Model for cascading failures in complex networks' [22] large, even if rare, cascades can be triggered in CEdS by small initial shocks when systems are composed of interconnected sub-systems or inter-dependent networks. Indeed, the emergent behaviour of CEdgSs arising from the interaction of the many sub-systems and components is highly non linear and impossible to predict sufficiently with the traditional tools, as it is stated by Calvano C. et al. in 'Systems engineering in an age of complexity' [4]. A perfect control of the system behaviour and safety, then, could probably never be obtained, but Complex Theory could help in increasing the level of control we have. This is specific for CEdgSs, differently from traditional

systems and SoS as explained by Sheard et al. in 'Principles of complex systems for systems engineering' [9]. The authors also argue that CEdgS stay at the border of the order-to-chaos spectrum. Modelling the complexity within engineered systems is then of fundamental importance. From a mathematical point of view, in [22] the authors present a simple model based on graph theory to simulate the dynamic cascade of effects that starting from a single node removal lead to macroscopic failures. From a more engineering perspective, instead, Punzo G et al. in 'Engineering Resilient Complex Systems: The Necessary Shift Toward Complexity Science' [23], state that the solution to the problem of the overarching growth of complexity in CEdgS can be achieved with the new research field of 'Design for Resilience'.

The concepts of 'Resilience' and 'Resilience Engineering' are relatively recent and derive from two decades of research. First scholars tried to formalise the definition of resilience and then they developed methods to model and quantify it within the design process of engineered systems. Attempts to define 'Resilience' can be found in 'Framework for analytical quantification of disaster resilience' by G. P. Cimellaro et al. [24], 'A review of definitions and measures of system resilience' by S. Hosseini et al. [25], 'Essential characteristics of resilience' by D. Woods et al. [26] and 'Essentials of resilience, revisited' by [27], D. Woods et al. The reader could find an overview of the 'Resilience Engineering' problem, current research directions and challenges in [23]. In this paper, an analogy with biological systems is used to define CEdS as 'living' and adapting systems that can change and react to inputs from the environment and also from internal unexpected events. To overcome the problem of an increasing complexity, the authors suggest an overarching graph theory approach on top of the engineering modelling and optimisation process to investigate the resilience of CEdS, including temporal evolution, resilience features, the management and decision layers, and the transparency of boundaries between interconnected systems. In 'A Review of Methods to Study Resilience of Complex Engineering and Engineered Systems' [28] it is recognised that uncertainty and interconnectedness are the main drivers of complexity in modern CEdgS and that a paradigm shift in the approach for the design is needed. Six methods, three each for uncertainty and interconnectedness, have been identified and presented. With reference to methods for UQ the authors suggest Bayesian Networks (BNs), Robust Bayesian Modelling for severe uncertainty where families of probability distributions are propagated and finally MDO techniques under uncertainty where decoupling methods are used to properly decompose CEdgS together with UQ methods. In addition, the paper also mentions three methods for modelling interconnectedness in CEdgS: Network Science, Cellular Automaton (CA), and Agent Based Modelling (ABM). Network Science helps understanding cascade effects and the relationships between local and global dynamics. CA combined with re-configurable platforms offer new possibilities for resilience design strategies. ABM integrates technical and social components in complex systems and allows for the explicit description of autonomous and heterogeneous facets. Multiple limitations of each method, including computational costs, scalability issues, data uncertainties, simplification challenges, and the difficulty of predicting human behaviour are finally presented.

2.5 Necessary Tools for Systems Engineering

Engineering complexity by optimising resilience requires the orchestration of many and diverse engineering and mathematical tools. This section serves as an introduction to the ones used in this thesis.

Network Theory

Networks representations of complex systems is an interesting research topic that spans in a wide range from pure mathematics to engineering approaches: Graph Theory, Network Theory, Complex System Theory, ... Complex System Theory is particularly interesting for the design of CEdgSs for it integrates network theory with system theory and it can deal with complexity in engineered systems. The base of all these tools is the mathematical study about graphs. A graph is an ordered pairs G=(V,E) comprising: (i) V, a set of vertices (also called nodes or points); (ii) $E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\}$, a set of edges (also called links or lines), which are unordered pairs of vertices (that is, an edge is associated with two distinct vertices). For an introduction to complex network, please refer to [52, 53]. In the following of the thesis a particular type of graph will also been considered: the multi-layer graph: a pair (G, C) where: (i) $G = \{G_\alpha; \alpha \in \{1, .., N_L\}\}$ is a family of directed and weighted graphs $G_\alpha = (X_\alpha, E_\alpha)$ with N_L the number of layers; (ii) $C = \{E_{\alpha\beta} \subset X_\alpha \times X_\beta; \alpha, \beta \in \{1, 2, ..., N_L\}, \beta = \alpha + 1\}$ is the set of interconnections between nodes of different layers. A survey of multi-layer network can be found in [95].

Catastrophe Theory

Scientists have found that there are two basic types of processes in nature: continuous and discontinuous. An example of a continuous process is the increase in temperature of a gas as it is heated. As one variable is changed at a constant rate (heat is added to the gas), a second variable also changes at a constant rate (the temperature of the gas increases). Because continuous processes are "smooth," they are relatively easy to predict. The branch of mathematics used to study continuous processes is called calculus and was developed by Isaac Newton (1642-1727) and Gottfried Leibniz (1646-1716) more than 300 years ago. Discontinuous processes are instead studied in Bifurcation and Catastrophe theories [44]. Catastrophe theory is a branch of Bifurcation Theory, initiated by the French mathematician Ren Thom in the 1960s and popularised by Christopher Zeeman in the 1970s. It studies and classifies phenomena characterised by sudden shifts in behaviour arising from small changes in circumstances. A brief introduction to the problem can be found in 'PhD mini course: introduction to bifurcation analysis' by VanVoorn [96] and in 'Dynamics and bifurcations of non-smooth systems: A survey' [45] by Makarenkov et al. The latter presents also some practical implementations of bifurcation models. Sharma et al. in 'Numerical continuation and bifurcation analysis in aircraft design: an industrial perspective' [97] show an application to a real engineering problem and presents progress on how bifurcation analysis can play a role as part of the design process for passenger aircraft. Other interesting works to approach the field are: 'Homoclinic and heteroclinic bifurcations in vector

fields' by Homburg et al. [98], 'Switching to nonhyperbolic cycles from codimension two bifurcations of equilibria of delay differential equations' by Bosschaert et al. [99], 'Hyperchaos in a spacecraft power system' by Li et al. [100]. An example of a discontinuous process involves an arched bridge to which more and more weight is added. At first, little effect is seen as the weight on the bridge is increasedthe bridge begins to bend almost imperceptibly. At a certain point, however, enough weight is added to the bridge that it collapses. A steady change in one variable (the amount of weight on the bridge) results in almost no change in a second variable (the shape of the bridge distorts slightly) followed by a sudden change to a very different state (the bridge collapses). A sudden change in a discontinuous process is called a catastrophe. In mathematics catastrophes can include sudden disasters, such as a bridge collapse or an earthquake, but they can also include much less dramatic events, such as the boiling of water or any phase transition phenomena. For processes involving four variables, Thom discovered that there are seven basic types of catastrophes. They are named for the shapes formed when their variables are graphed: fold, cusp, swallowtail, butterfly, wave, hair, and fountain. In practical terms, the theory examine the critical points or equilibrium states of a dynamical system as a parameter μ is varied:

$$\dot{x} = h_\mu(x). \tag{2.1}$$

They investigate how the system transitions from one stable state to another or undergoes qualitative changes in its behaviour. In other words, it focuses on the analysis of the structural changes that occur in a system when one or more parameters cross certain critical values. Bifurcations can lead to the emergence of new stable states, limit cycles, chaos, or other types of dynamic behaviour including the switch between attraction and repulsion.

Stochastic Process

A stochastic or random process can be defined as a collection of random variables that is indexed by some mathematical set, called the index set, which has usually the interpretation of time:

$$X(t), t \in T \subseteq \mathbb{R}_+ \tag{2.2}$$

Stochastic processes can be grouped into various categories, which include random walks, martingales, Markov processes, Lvy processes, Gaussian processes, random fields, renewal processes, and branching processes to cite a few. A classification can also be made based on the nature of the state space and on the time set: discrete-time or continuous-time stochastic processes on one side and discrete, integer-valued stochastic process or real-valued stochastic process on the other side.

Common examples of stochastic processes are: Bernulli Process, Random walks, Poisson Processes and Markov chain. This modelling frameworks have demonstrated to be useful in many fields: the Wiener process (or Brownian motion process), was used by Louis Bachelier to study price changes on the Paris Bourse, and the Poisson process, was used by A. K. Erlang to study the number of phone calls occurring in a certain period of time.

Chapter 8 will build on top of this tools and, in particular, it includes a resilience model based on the Homogeneous Continuous-Time Markov Chain (HCTMC) which is a particular case of Markov Process. A Markov Process is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. This is a property called 'memorylessness'. The model is named after the Russian mathematician Andrey Markov. Markov Chain is then the particular case of Markov Process with a countable state space. Markov Chain are usually classified in a Discrete-Time Markov Chain (DTMC) if discrete-time and Continuous-Time Markov Chain (CTMC) is continuous-time process. The evolution of the system states through different points is called 'transition' and it follows a specified probability distribution defined in the transition matrix **P**. The HCTMC have finally the transition matrix **P** that is the same after each time step.

Modelling

A model in science and engineering is anything used as a representation of a law, theory, event, object or system. It describes the real system including only the essential features. Modelling is then the process of abstraction from reality and of definition, by means of a particular language, of the model [30].

Real systems can be translated to an infinite number of virtual or physical models since they can be defined at different levels of abstraction and fidelity and with different modelling approaches and languages.

The modelling process is composed of a few important steps: the identification of the real system under interest, its boundaries and which system's property can be neglected, the selection of a modelling paradigm approach and finally a modelling language.

In general, models can be classified based on their purpose, that is the final goal for which the model is developed. A model can be used for prediction, description, design/optimisation, etc. Models can also be classified based on the used substrate: a model can be for example conceptual, mathematical, or physical. Restricting our attention to mathematical models, the ones used in the thesis, we can further consider other categories. (i) **Type**: *analytical* if it is based on the pure application of a theory, *numerical* if it includes also numerical approximations and simplifications, *empirical* if it contains parameters that must be quantified by experiment or systematic observation. (ii) **Mathematical framework**: differential equation, finite difference equation, combinatorial theory, probability theory, game theory, game trees or payoff matrices, ... (iii) **Linear or Non-linear**: the model can be linear or non-linear with respect to the independent variables. (iv) **Static or Dynamic**: static models refers to steady state systems or in general to systems that don't change their properties in time. Models are defined dynamic otherwise. The latter, in particular, are the base of the simulation process. (v) **System Knowledge**: if both all the input parameters of the model and the set of operations to calculate the model's output are known, it is defined a *white box*. Otherwise it is define as *black box*. Sometimes also the definition of *grey box* is used. (vi) **Discrete or Continuous**: this refers to the independent variables of the model. (vii) **Uncertainty**: a model can be *deterministic* if it does not

include uncertainty. A deterministic model gives always the same result. Models can also include uncertainty that can be treated with many different approaches. An example of the latter is given by the *stochastic models*. (viii) **Modelling and Simulation Paradigm**: the analysis of a dynamic model is made by means of simulations. The most common model paradigms that allows for simulation are: ABM, Discrete Event Simulation (DES), Dynamic Systems (DS) and SD and CA.

The simulation process of dynamic models allows us to study the system's time evolution. A clear comparison of ABM, DES, DS and SD is in [101]. Many other approaches however exist. (i) **System Dynamics**, SD, was developed by the electrical engineer Jay W. Forrester in the 1950s. It is based on the concepts of stocks (e.g. of material, knowledge, people, money), flows, feedback loops and time delay. SD was first applied only to management problems but the range of applications soon expanded also to urban, social, ecological and other types of systems. (ii) **Dynamic Systems**, DS, is actually the ancestor of System Dynamics. It is similar to SD but for historic reason, it uses a different set of tools. It is applied for the solution of physical and engineering problems only. (iii) **Discrete Event Simulation**, DES, roots to 1960s with Geoffrey Gordon at IBM. It is based on the concepts of entities, resources and block charts describing entity flow and resource sharing. Each event occurs at a particular instant in time and triggers changes a change in the state of the system. Between consecutive events, no change in the system is assumed to occur. (iv) **Agent-based Modelling**, ABM, is a distributed approach that simulates the behaviour of individual agents in a system, and the interactions between those agents. It is often used to study complex systems in which individual agents have a degree of autonomy, such as social or economic systems. The agents have an individual capacity of sensing the environment, the ability to learn and react. (v) **Cellular Automaton**, CA, was started by John von Newman, and is the ancestor of ABM. A CA is a modelling approach in which a system is represented as a grid of cells, each of which can be in one of several states. The states of the cells change over time according to a set of rules, which can be simple or complex. CA are often used to model physical or biological systems, such as fluid dynamics or ecological systems. (vi) **Game Theory** is finally a mathematical framework for analysing the behaviour of individuals or groups in strategic situations. It involves modelling the decisions and interactions of players in a game and predicting the outcomes based on the assumptions of the game. Game theory is often used in economics, political science, and other fields to model decision-making and strategic behaviour.

Many specialised modelling languages have been finally developed. The main classification is between **Graphical Languages** like Bond Graph, Unified Modelling Language (UML), Systems Modelling Language (SysML) and **Textual Languages** such as formal codes based on standardised keywords and natural language terms. An important classification about specialised modelling languages refers to the possibility to apply a *causality* or *a-causality* approach as explained in [102]. In summary, we can say that in the **causal notation**, causal models have the outputs explicitly expressed in terms of the inputs; i.e., the direction of information flow is manifest. Concretely, such models are often represented as block diagrams. An examples is SimuLink. In the **declarative or a-causal notation**, a-causal models are expressed in terms of undi-

rected equations. This makes them much more reusable and composable, addressing some of the challenges of large-scale modelling and simulation. In alternative, general-purposed languages can be used to manually implement the physical equations of the system through a Equation Based Modelling (EqBM) approach.

3

Evidence-Based Uncertainty Quantification

" Is there more to Uncertainty than
some probability theorists might have us believe? " [46]

– George J. Klir

Reasoning about real problems means reasoning under uncertainty since every description of the real world is necessarily incomplete and affected by a degree of uncertainty. Within the field of MBSE it is then necessary to understand the different sources and types of uncertainty that affect the engineered system under design and the whole design process, to properly model them and to quantify their effect on the final output. Also, since the classical probability framework has been demonstrated [46–48] to capture only part of the multidimensional concept of uncertainty, IP theories can be exploited to better model and understand reality.

This chapter introduces the concept of uncertainty, of its modelling and explains why DST has been selected as the framework for UQ in the proposed MBSE methods for EBRO and EBORe presented in the thesis. The chapter then moves to the presentation of different algorithmic methodologies that are used in the literature to solve the UQ problem.

This chapter is linked to Chapter 6 which presents a modelling framework for CEdgS under uncertainty and to Chapter 7 which explains novel algorithmic approaches for Belief and Plausibility approximation for computational complexity reduction. It is also linked to all the MBSE applications presented in the following chapters Chapters 7 to 9.

In particular, Section 3.1 introduces the UQ problem. It presents the different interpretations and types of uncertainties, lists a number of examples of uncertainty manifestations, presents the two main problems in which UQ is involved and the most important mathematical theories. Section 3.2 is specifically dedicated to DST, its measures of information, its advantages and disadvantages. Section 3.3 are then focused on the algorithmic approaches for the solution of the UQ problem for the general probabilistic approaches and for DST.

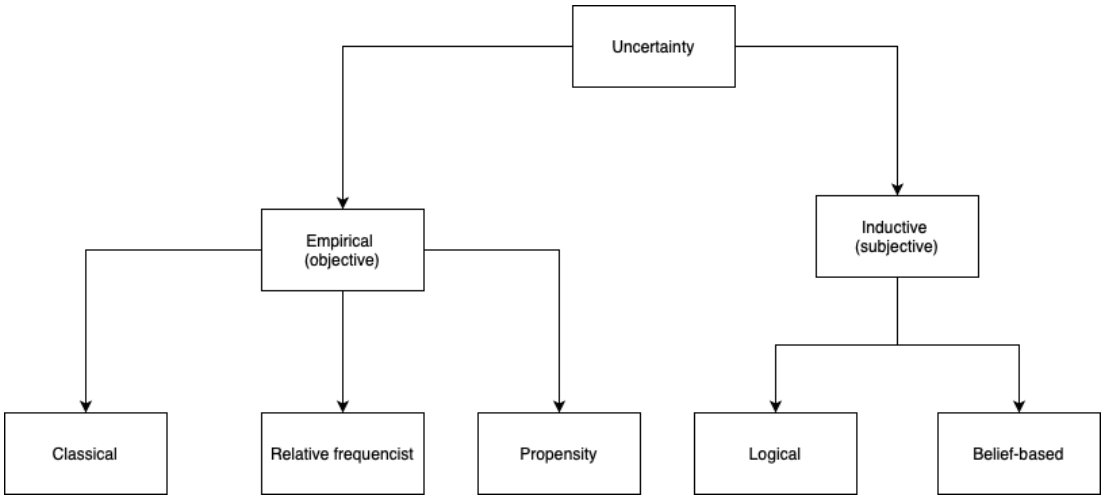


Figure 3.1: Main Uncertainty Interpretations

3.1 What Uncertainty Is and How to Live With It

The concept of uncertainty has been evolving, strictly interconnected to the ones of knowledge, information and probability, along the centuries. This section gives a brief introduction to most important aspects we can think about uncertainty. First, from the highest level, we give the different interpretations which relates to the intuition at the beginning of the development of the uncertainty concepts. The two most common categories of uncertainty are then presented: aleatory (or irreducible) and epistemic (or reducible). Some important examples of how this concepts manifests in real life are then given. The section then gives a comprehensive, even if not complete, list of mathematical theories to model uncertainty. Finally there is a brief description of the problems related to uncertainty.

The main distinction of uncertainty interpretations is between **empirical** (or objective) and **inductive** (or subjective) approaches as [103] and here summarised in Fig. 3.1. The former case can be further divided in three sub-classes. **Classical** interpretation is based on the knowledge of all the possible outcomes and on symmetry. **Frequency** interpretation is based on the repetition of a large number of experiments. **Propensity** interpretation is a natural inclination or preference toward a particular state. The inductive approach can instead be divided in two main sub-groups. **Logical** interpretation is based on reasoning rules (as the inference rules). **Belief-based** interpretation is based on the human state of mind and convictions.

Different types of uncertainty can also be identified and categorised based on the physical properties of the available information. An important classification divides uncertainties into aleatory and epistemic [49]. **Aleatory** uncertainties are non-reducible uncertainties that depend on the very nature of the phenomenon under investigation. They can generally be captured by well-defined probability distributions as one can apply a frequentist approach. **Epistemic** uncertainties are reducible uncertainties and are due to a lack of knowledge. Generally, they cannot be quantified with a well-defined

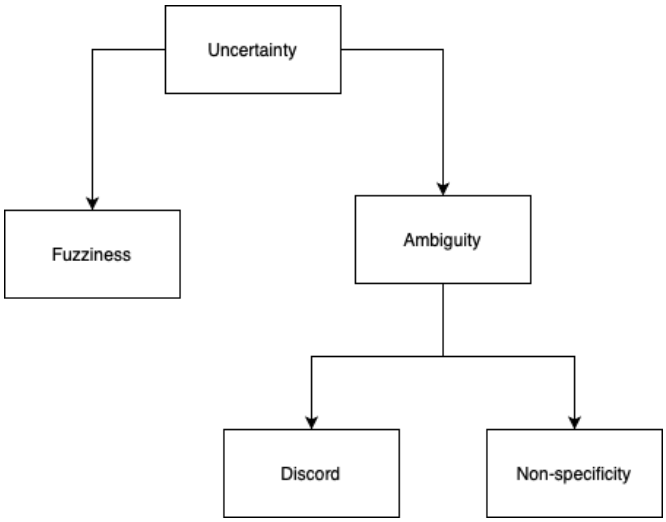


Figure 3.2: Different interpretations of uncertainty

probability distribution and a more subjectivist approach is required. Two classes: a lack of knowledge of the distribution of the stochastic variables or a lack of knowledge of the model used to represent the phenomenon under investigation.

Uncertainty is, however, a multidimensional concept and different taxonomies have been developed in the last years by many scholars in order to include its different and multiple facets: confusion, error, irrelevance, distortion, incompleteness, undecidability, inaccuracy, probability, ambiguity, fuzziness, non-specificity, vagueness, ignorance, conflict, imprecision, ... [104]. Of particular importance is the type taxonomy of *uncertainty-based information* of Generalised Information Theory (GIT) given by Klir [47] and represented in Fig. 3.2, where information and uncertainty are complementary aspects. An other classification is from Smets [105] who builds a larger typology of *imperfection of information* where imperfection has three dimensions: imprecision, inconsistency and uncertainty.

Uncertainty can then manifests itself in many and different circumstances. Some common examples are in the following. **Structural** (or model) uncertainty is a form of epistemic uncertainty on our ability to correctly model natural phenomena, systems or processes. If we accept that the only exact model of Nature is Nature itself, we also need to accept that every mathematical model is incomplete. One can then use an incomplete (and often much simpler and tractable) model and account for the missing components through some model uncertainty. **Experimental** uncertainty is generally aleatory but if one considers the uncertainty associated with measurements it can be considered epistemic as it incorporates the possible lack of knowledge on the performance of the sensor. Furthermore, a lack of measurements is in itself an epistemic uncertainty. When this uncertainty is aleatory, it is probably the easiest to understand and model, if enough data are available on the exact repeatability of measurements. **Geometric** uncertainty is a form of aleatory uncertainty on the exact repeatability of the manufacturing of parts and systems. **Parameter** uncertainty can be either aleatory or epistemic and refers to the variability of model parameters and boundary conditions. **Numerical** (or

algorithmic) uncertainty, also known as numerical errors, refers to different types of uncertainty related to each particular numerical scheme, and to the machine precision (including clock drifts). **Human** uncertainty is difficult to capture as it has both aleatory and epistemic elements and is dependent on our conscious and unconscious decisions and reactions. It includes the possible variability of goals and requirements due to human decisions.

The concept of uncertainty has been studied for centuries, but the development of formal theories and methods for dealing with uncertainty is a more recent phenomenon. Here is a brief history of some key milestones in the development of uncertainty theory:

- **Probability Theory** [106]. Probability theory dates back to the 17th century with the work of Blaise Pascal and Pierre de Fermat on gambling problems. It was later formalised by mathematicians such as Jacob Bernoulli, Abraham de Moivre, Kolmogorov and Thomas Bayes.

- **Fuzzy Set Theory** [107]. Fuzzy set theory was developed in the mid-20th century by Lotfi Zadeh, who proposed the idea of representing uncertainty using fuzzy sets, which allow for partial membership in a set. Fuzzy set theory has since been used in a wide range of applications, including control systems, decision-making, and artificial intelligence. Fuzzy sets theory is well suited to deal with vague information.

- **Dempster-Shafer Theory of Evidence**: Evidence Theory, also known as belief theory and DST, was developed in the 1960s by Arthur Dempster [50] and Glenn Shafer [51] as a way of dealing with uncertain and conflicting evidence. It extends probability theory by allowing for representing uncertain information using belief functions, which can be combined using the Dempster-Shafer rule of combination and further rules developed in the following years. Evidence theory has great capacity in ignorance modelisation.

- **Possibility Theory** [107]. Possibility theory, which was developed by Zadeh in the 1970s, provides a way of representing uncertain information using possibility measures, which represent the degree to which a proposition is possible. Possibility theory has been used in applications such as decision-making and pattern recognition. Possibility theory is appropriate for incomplete information.

- **Interval Analysis**: Interval analysis, which provides a way of representing and propagating uncertainty using interval arithmetic, was developed in the 1960s and 1970s by mathematicians such as Ramon Moore and Charles Steward. Interval analysis has been used in applications such as control systems and scientific computing.

- **Bayesian Inference**: Bayesian inference, which provides a way of updating probabilities based on new evidence, has been developed over the past few centuries, but has seen a resurgence in popularity in recent years due to advances in computational methods for inference.

These are just a few of the many developments in uncertainty theory over the years, and the field continues to evolve with new methods and applications.

Since the last decade of the 20-th century there has been then an effort to harmonise these different theories of uncertainty in the GIT [47,48]. In GIT uncertainty is viewed as a manifestation of information deficiency, while information is anything capable of reducing the uncertainty. All the approaches considered in GIT were developed after two important theories came up in the 20th century: the Theory of Capacities (1953) by Gustave Choquet and the *fuzzy* set theory introduced in 1965 by Lotfi Zadeh. The first extended the classical and additive concept of measure to the monotone measure, while the second expanded the classical set theory. As described in [47, 48] all uncertainty theories can be sorted and related based on these two aspects: the measure used to quantify uncertainty and the type of set implemented.

There are finally two major types of problems in UQ that people are interested to solve. In **Forward Propagation** of uncertainty, the various sources of uncertainty are propagated through the model to predict the overall uncertainty in the system response In **Inverse Propagation** there is an assessment of model uncertainty and parameter uncertainty: the model parameters are calibrated simultaneously using test data.

3.2 Evidence Theory

DST [50, 51] can be considered as a generalisation of the classical probability theory since it allows one to treat both aleatory and epistemic uncertainty with partial information and lack of knowledge.

Formally, an application of DST can be defined by the triple

$$(\Theta, \Omega, \mathrm{m}) \tag{3.1}$$

The set Θ is called *Frame of Discernment* and it is the set of all the mutually exclusive and collectively exhaustive elementary events (or hypothesis) θ_i, $i = 1, ..., |\Theta|$:

$$\Theta = \left\{\theta_1, \theta_2, ..., \theta_i, ..., \theta_{|\Theta|}\right\} \tag{3.2}$$

All the possible events (or hypotheses) could be overlapping or nested, but in the *frame of discernment* only the finest division of them is considered. Θ corresponds to the finite sample space in Probability Theory.

Ω is a countable collection of subsets of Θ. It corresponds to the σ-algebra for the Probability Theory, but it is less restrictive. Usually Ω is identified as the *power set* $2^\Theta = (\Theta, \cup)$ given by all possible combinations of the elements of Θ:

$$\Omega = 2^\Theta = \left\{\emptyset, \{\theta_1\}, ..., \left\{\theta_{|\Theta|}\right\}, \{\theta_1, \theta_2\}, ..., \{\theta_1, \theta_2, ..., \theta_i\}, ..., \{\theta_1, \theta_3\}, \Theta\right\} \tag{3.3}$$

where the generic element $\omega = \{\theta_1, ..., \theta_j\}$ of $\Omega = 2^\Theta$ is a proposition that states the truth of only one of the events $\theta_1, ..., \theta_j$ without specifying which one.

m is weight a function called basic probability assignment (bpa) defined on the elements of Ω.

$$m : \Omega \to [0, 1]. \tag{3.4}$$

Given $\epsilon \in \Omega$, the m is a number characterising the amount of likelihood that can be assigned to ϵ. It has to satisfy the following conditions:

$$m(\omega) \geq 0, \forall \omega \in \Omega \tag{3.5}$$
$$m(\omega) = 0, \forall \omega \notin \Omega \tag{3.6}$$
$$m(\emptyset) = 0 \tag{3.7}$$
$$\sum_{A \in 2^{\Theta}} m(A) = 1 \tag{3.8}$$

m corresponds to the *pmf* in the classical probability theory.

Each subset of the *power set* 2^{Θ} with a non-zero bpa is called a Focal Element (FE) and the pair $\langle F, m \rangle$, where F is the set of all FEs and m the corresponding bpas, is called *Body of Evidence*.

3.2.1 Belief and Plausibility

This section is dedicated to the definition of Belief (Bel) and Plausibility (Pl) which are the two uncertainty measures in DST that generalise probability being respectively its lower and upper bound. The gap between them, called Evidential Interval (EI) $[Bel \quad Pl]$ is a measure of the degree of ignorance on the probability of a realisation of a general event x. When enough information is available, the distribution of x is known and all FEs collapse to singletons converging to the state $Bel = Pl = P$.

Given A a subset of the power set $A \subseteq \Omega = 2^{\Theta}$ and ω_i the generic i-th FE inside $\langle F, m \rangle$ the total degree of Bel on A can be computed by collecting all the pieces of evidence that fully support that statement A:

$$Bel(A) = \sum_{\omega_i \subseteq A} m(\omega_i) \tag{3.9}$$

The total degree of Pl can be computed by collecting all the pieces of evidence that either fully or partially support that statement A.

$$Pl(A) = \sum_{\omega_i \bigcap A \neq \emptyset} m(\omega_i) \tag{3.10}$$

From Eq. (3.9) one can see that the Bel function is the sum of all the pieces of evidence that completely support the statement $x \in A$, whereas the Pl function is the sum of all the pieces of evidence that partially support the statement $x \in A$. This means that $m(\omega_i)$ is added to Bel only if all possible realisations of $x \in \omega_i$ belong to A, on the contrary $m(\omega_i)$ is added to Pl if at least one realisation of $x \in \omega_i$ belongs to A.

The functions Bel and Pl follows a less restrictive version of Kolmogorov's axioms:

1. $Bel(\emptyset) = 0$;
2. $Bel(U) = 1$;

3. For every positive integer n and every collection $\theta_1, ...\theta_n$ of subsets of Θ:

$$Bel(\theta_1 \cup ... \cup \theta_n) \geq \sum_i Bel(\theta_i) - \sum_{i<j} Bel(\theta_i \cap \theta_j) + -... + (-1)^{n+1} Bel(\theta_1 \cap ... \cap \theta_n)$$

and

1. $Pl(\emptyset) = 0$;
2. $Pl(U) = 1$;
3. For every positive integer n and every collection $\theta_1, ..., \theta_n$ of subsets of Θ:

$$Pl(\theta_1 \cap ... \cap \theta_n) \geq \sum_i Bel(\theta_i) - \sum_{i<j} Pl(\theta_i \cup \theta_j) + -... + (-1)^{n+1} Pl(\theta_1 \cup ... \cup \theta_n)$$

Condition (3.) for both Bel and Pl are a further generalisation with respect to the classical probability theory since the two measures are defined as monotonic non-additive.

A further generalisation of the classical probability theory is related to the concept of duality:

$$Pl(\neg A) = 1 - Bel(\overline{A}) \tag{3.11}$$

with $\overline{A}$ the complement to A.

3.2.2 Evidence Framework for System Engineering

This section is dedicated to the combination of DST and MBSE. In the design process of an engineered system, within the framework of MBSE, a model f is used to quantify one important KPI. f depends on both a vector of design $\mathbf{d}$ and uncertain $\mathbf{u}$ variables as in the following definition:

$$f(\mathbf{d}, \mathbf{u}) : D \times U \subseteq \mathbb{R}^{m+n} \rightarrow \mathbb{R} \tag{3.12}$$

where D is the design space for the decision or design parameters $\mathbf{d}$, of dimension m, and $U = 2^{\Theta}$ the event space for the uncertain parameters $\mathbf{u}$, of dimension n, that we call the *Uncertain Space*.

We are then interested in the amount of evidence associated with the event $f(\mathbf{d}^*, \mathbf{u}) \in B$, for a fixed $\mathbf{d}^*$ and we want to quantify the level of uncertainty on the set A:

$$A = \{\mathbf{u} \in U | f(\mathbf{d}^*, \mathbf{u}) \in B\} \tag{3.13}$$

In other words, we are aware of the probability space $(\Theta_A, \Omega_A, \mathrm{m})$ and we want to apply a forward uncertainty propagation through the system model f to understand the effect of uncertainty on the model output.

With reference to Eq. (3.13) we want finally to compute the cumulative Belief and Plausibility associated with the event A:

$$Bel(A) = \sum_{\omega_i \subset \Omega, \omega_i \in A} bpa(\omega_i), \tag{3.14}$$

$$Pl(A) = \sum_{\omega_i \cap \Omega \neq 0, \omega_i \in A} bpa(\omega_i). \tag{3.15}$$

3.2.3 Advantages and Disadvantages of Evidence Theory

A discussion on the main advantages and disadvantages of DST can be found in [108] by Liu et al. Differently from classical probability theory and from the Bayesian approach, the epistemic framework in DST includes the concept of ignorance. The evidence framework makes it possible, indeed, to profess zero degree of belief for any proper subset of a set of possibilities. For instance, evidence on the event A or B does not imply/require information on both events A and B. Similarly, the knowledge of an event does not imply knowledge of its opposite (for the probability theory P(A)=1P(A)).

On the other hand, the Bayesian theory cannot deal so readily with the representation of ignorance. The basic difficulty is that the theory cannot distinguish between lack of belief and disbelief. Also Probability Theory (PT) has this limitation: it assigns equal probability to all the possible events when no information is available, what is called the Principle of Indifference or of non-sufficient reason. Thus, PT does not make any distinction between randomness and ignorance.

In traditional probability theory evidence is related only to a single event, instead in DST it is associated with a set of events and the mass assigned to a general set A gives no information about the evidence associated with the subsets of A.

On the other side, the main problem of UQ with DST, and with IP in general, is its computational complexity. This is a limitation also for DST where the cost for exact quantification of uncertainty grows exponentially with the uncertainty space. Indeed, the definition of Bel and Pl requires searching for extreme events in each FE.

The solution of Eq. (3.18) is far from trivial. In fact, the computation of Bel presents two major difficulties:

- In order for a FE ω to be included in the calculation of the belief, the following condition must be true:
$$\max_{\mathbf{u} \in \omega} f \leq \nu \tag{3.16}$$
which implies solving a number of (global) maximisation problems equal to the number of FE.

- Because FEs can be either fully included or fully excluded from the calculation of $Belief$, the function $Bel(\mathbf{d})$ is generally discontinuous, non-differentiable and presents plateaus that make it unsuitable for a gradient methods.

This explains the importance for developing algorithmic procedures to reduce the computational cost in order to apply DST within MBSE.

3.3 Algorithms for Uncertainty Quantification

This section presents a list of the most important approaches for forward UQ. The problem refers to the propagation of uncertainty through a general black-box-type performance function denoted as

$$y = g(\mathbf{x}) \tag{3.17}$$

where $\mathbf{x}$ is the n-dimensional vector of uncertain variables.

In the general probabilistic approach, where $\mathbf{x}$ can be described by the joint Probability Density Function (PDF) $f_X(\mathbf{x})$, there are basically six categories of methods for uncertainty propagation as described in [109]. The first category includes simulation-based methods like Monte Carlo simulations, importance sampling, adaptive sampling, etc. [110]. The second category refers to the general surrogate-based methods. In a non-intrusive approach, a surrogate model is learnt in order to replace the experiment or the simulation with a cheap and fast approximation. Surrogate-based methods can also be employed in a fully Bayesian fashion. This approach has proven particularly powerful when the cost of sampling, e.g. computationally expensive simulations, is prohibitively high. Then there are local expansion-based methods. Examples are Taylor series and perturbation method. These methods have advantages when dealing with relatively small input variability and outputs that don't express high non-linearity. The fourth category is given by functional expansion-based methods: Neumann expansion, orthogonal or KarhunenLoeve expansion (KLE), with Polynomial Chaos Expansion (PCE) [111] and wavelet expansions as special cases. There are then the Most probable point (MPP) based methods as First-Order Reliability Method (FORM) and Second-Order Reliability Method (SORM). The last category is numerical integration-based methods: Full Factorial Numerical Integration (FFNI) and Dimension Reduction (DR).

Focusing instead on Evidence Theory approaches for UQ, the general problem Eq. (3.17) translates to:

$$Bel(f(\mathbf{u}) \leq \nu) \tag{3.18}$$

for a specific ν or $\forall \nu \in [min_u(f), max_u(f)]$ where $\mathbf{u}$ is defined as in Section 3.2.

In the literature one can find a handful of sampling-based methods to compute and estimation of *Belief* (and *Plausibility*) [112–115]. As a representative example, in [113], the authors suggest the use of the density function:

$$d_j(u_j) = \sum_{k=1}^{n(j)} \delta(u_j | I_{jk}) bpa_{jk}(I_{jk}) / (b_{jk} - a_{jk}) \tag{3.19}$$

for the j-th dimension of the uncertain space, if the intervals for the uncertain parameters are given in the form $I_{jk} = \{u_j : a_{ik} \leq u_j \leq b_{jk}\}$. Here $n(j)$ is the number of intervals in the j-th dimension and

$$\delta(x_j | I_{jk}) = \begin{cases} 1 & \text{if } x_j \in I_{jk} \\ 0 & \text{otherwise;} \end{cases} \tag{3.20}$$

then the sampling distribution is:

$$d(\mathbf{x}) = \prod_{j=1}^{nU} d_j(u_j) \tag{3.21}$$

Distribution (3.21) explores adequately the uncertain space giving more importance to the FEs with higher bpa and then sampling uniformly inside them. Samples can be generated with a Latin Hypercube Sampling (LHS) and propagated through the system model in order to build a response surface, for example through a non-parametric regression model, that can either directly approximate the Bel or the quantity of interest from which one can calculate the Bel.

Sampling-based approaches are potentially efficient but produce an approximation that can be significantly poor. Furthermore, this type of approximation provides estimated $Belief$ values that are always better (more optimistic) that the actual ones, leading to an overconfidence in the realisation of an event. Possible mitigation of this problem was recently proposed in [116] to address the solution of optimal control problem under epistemic uncertainty. The value of the $Belief$ was approximated with the surrogate of a weighted integral obtained by sampling the space of the FEs. The integral was elevated to an exponent factor k, the higher k the more the integral was resembling the actual Bel. Furthermore, the surrogate was periodically updated to identify the threshold values where the approximation was the closest to the true $Belief$. These two improvements allows one to insert selected values of the surrogate in the optimisation loop and improve the Bel at a discrete number of thresholds ν.

If sampling is used to build a surrogate of the quantity of interest, the computational cost due to the maximisation over all the FEs is only partially mitigated, as the cost of each optimisation is reduced but the number of optimisations might remain very high and could still scale exponentially with the number of dimensions of the uncertainty space U.

Another class of approaches then tries to reduce the number of FE upfront through a dimentionality reduction methods. The general idea is to define criteria to sort the FEs by their importance and then approximate the m-function in Eq. (3.4) to a m'-function with a lower number of FEs. A few examples are presented in the following.

The Bayesian approximation proposed by Voorbraak's [117] produces a discrete probability distribution where the new mass function m' considers only the singleton subsets θ_i in the Power Set 2^Θ. The Consonant approximations from Dubois and Prade [118] implement and approximation through the use of consonant sets, in the sens of randomness. This results to be equivalent to a fuzzy approach which finally yields to a reduction of computational cost. The k-l-x method proposed by Tessem [119] uses the bpa's as sorting criterion. The approximating m'=m_{klx} then includes only the p FEs with higher bpa, where $k < p < l$, and such that the sum of the masses of the deleted FEs is less than x. A normalisation method is finally used to redistribute the total mass of the deleted FEs to the remaining ones. The *summarisation method* takes the first p FEs with the highest bpa, as in [119], and lumps together all the remaining ones in a single FE with a bpa that is the sum of their bpas. The *D1 method* [120] beside the criteria of *mass* introduces also the *cardinality*. The *Batch approximation method*

and *Iterative approximation method* [121] suggest, instead, that *mass* and *cardinality* are not sufficient to discriminate which FE to take and which one to discard. Then in the paper a *non-redundancy* measure is presented based on the definition of distance between two FEs as proposed by Denœux in [122]. Other heuristic methods can be found in [123]. A further approach in this category is presented as a contribution of the thesis in Chapter 7 where a reduction of the number of FEs is derived by a system decomposition methodology. A second method is presented instead in Appendix C.

4

Optimisation Under Uncertainty

“ There ain’t no such thing as a free lunch ”

– Unknown

“ In a predestinate world, decision would be illusory;
in a world of a perfect foreknowledge, empty;
in a world without natural order, powerless.
Our intuitive attitude to life implies
non-illusory, non-empty, non-powerless decision...
Since decision in this sense excludes both
perfect foresight and anarchy in nature,
it must be defined as choice in face of bounded uncertainty. ”

– Shackle, 1961

In order to apply an optimisation-based MBSE approach to the design of a system, the engineering problem needs to be translated in the optimisation problem which can then be solved with the use of a suitable optimisation solver. The whole process requires the identification of the system under analysis, the formulation of an adequate system’s model, the identification of possible sources and types of uncertainty affecting the system and involved in the design process, the identification of the set of optimisation variables and constant problem parameters, and the definition of objective and constraint functions. As stated in the 1997 in the No Free Lunch (NFL) Theorems formulated by David Wolpert and William Macready in [124] all the algorithms are equivalent if averaged over all possible problems. In other words, the perfect optimisation solver does not exist. It is therefore important to be able to develop or choose the appropriate one based on the specific characteristics of the problem under analysis.

This chapter introduces to the field of global optimisation by giving fundamental definitions and classifications of both problem formulations and algorithmic solvers. Particularly important for the following of the thesis are: meta-heuristic approaches, min-max problem, OUU with uncertainty modelled through DST framework, MDO and SDO problems, multi-objective formulation and constraint handling. The interested reader can find a more detailed introduction to optimisation in [33–35].

The chapter is linked to Part II where the thesis’ contribution in the field of algorithmic optimisation is presented. In particular, Section 4.4 introduces to the constraint

min-max problem that is then developed in Chapter 5. Section 4.5 defines the SDO problem for which the thesis contribution is in Chapters 6 and 9. Section 4.2.1 presents the Memetic Algorithms (MA) strategy that is further developed in Chapter 5. The chapter is finally important for Part III where the thesis' contribution about OUU with DST and SDO is presented which uses the SDO model presented in Chapter 6.

Section 4.1 gives the general optimisation problem formulation. Section 4.2 instead presents types and classifications of algorithmic solvers. Section 4.3 focuses on OUU and in particular with DST. Section 4.4 presents the min-max problem. Section 4.5 introduces the MDO. Section 4.6 is dedicated to the MOP with particular emphasis on the scalarisation approaches. Section 4.7 is finally dedicated to the methods for handling constraints.

4.1 Problem Formulation and Classification

A generic optimisation problem assumes the following mathematical formulation:

$$\begin{aligned} &\text{minimise}_{\mathbf{x}\in\mathbb{X}} \ \mathbf{f}(\mathbf{x}) = [f_1, f_2, ..., f_m]^T \\ &\text{subject to} \quad c_i(\mathbf{x}) \leq 0, \ \ i = 1, ..., n \\ &\qquad\qquad\ \ g_j(\mathbf{x}) = 0, \ \ j = 1, ..., p \end{aligned} \tag{4.1}$$

with $\mathbf{f}$ the vector of objective functions, $\mathbf{c}$ the vector of inequality constraint functions and $\mathbf{g}$ the vector of equality constraint functions, $\mathbb{X} \subset \mathbb{R}^n$ the parameter space, $\{m, n, p\} \in \mathbb{N}$ and $\mathbb{Y} = \{\mathbf{f}(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \mathbb{X}, \ g_j(\mathbf{x}) \leq 0, \ j = 1, ..., n\}$ the feasible objective space. Eq. (4.1), in particular, represents a CMOP and is the most generic representation from which all the particular cases described in the following can be derived.

Depending on the characteristics of the objective and constraint functions and of the search space, the optimisation problems can be classified within many categories. (i) **Local or Global**. A multi-modal problem has multiple local solutions corresponding to different vectors of $\mathbf{x}$. On the other side, a uni-modal problem has only one local solution which corresponds to the global one. Examples of the latter are: Linear Programming (LP) and convex programming problems. (ii) **Single, Multi and Many Objective**. With reference to Eq. (4.1), Single-Objective Problem (SOP) have $m = 1$, MOP have $1 \leq m \leq 3$ and Many-Objective Problems (MaOP) have $m > 3$. (iii) **Constrained or Unconstrained**. With reference to Eq. (4.1), constrained problems have $n > 0$ and/or $p > 0$ while unconstrained problem have $m = 0$ and $p = 0$. Constraint problems and Multi-objective problems are two alternative representations of the same problem linked by a adequate transformation. (iv) **Continuous or Discrete**. A continuous problem is defined for $\mathbf{x} \in D \subseteq \mathbb{R}^n$. A discrete problem instead, has $\mathbf{x}$ defined in a discrete set. Examples of the latter are: *Integer Programming* when $\mathbf{x} \in D \subseteq \mathbb{Z}^n$, *Binary Programming* when $\mathbf{x} \in D = \{0, 1\}^n$ and *Combinatorial Optimisation* where the discrete set is a set of objects, or combinatorial structures, such as assignments, combinations, routes, schedules, or sequences. *Mixed-Integer Optimisation Problems* are also important: $\mathbf{x} = (\mathbf{x}_r, \mathbf{x}_d) \in D \subseteq \mathbb{R}^{n_r} \times \mathbb{Z}^{n_d}$, with $n_r + n_d = n$. (v) **Linear or Non Linear**. When all objectives and constraint functions are linear functions of

x the problem is defined as LP otherwise it is a Nonlinear Programming (NLP). A particular case of the latter is Quadratic Programming (QP). Integer Linear Programming (ILP) combines the last two classifications and is an example that shows that they represent different aspects of the problem and are usually combined. (vi) **Gray or Black Box**: sometimes the model under optimisation is known and it is possible to apply some decomposition techniques (grey box). Other times it is just a black box. (vii) **Number of Levels**. Often, an optimisation problem has a hierarchical structure: it is described by multiple optimisation tasks such that only an optimal solution to one of them may be a feasible candidate for the others. We can then distinguish between single-layer (standard problems), bi-level [125] (min-max, leader-follower, ...) and multi-layer optimisation problems. (viii) **Uncertainty Handling**. An important distinction is between deterministic formulations and problems that require also UQ. Uncertainty comes in different shapes and can be modelled in different ways. (ix) **Number of disciplines**. Sometimes, designers have to incorporate many relevant disciplines simultaneously. With reference to this aspect, it is possible to define MDO (or UMDO with the inclusion of UQ) when there are many fields weakly coupled and SDO when, on the opposite, there are fewer but stronger coupled fields.

A specific nomenclature exists with reference to constraint handling and the presence of the objective function as summarised in Table 4.1. (i) A Constrained Optimsation Problem (COP) has one or more objective functions and one or more constraint functions. (ii) A Constraint Satisfaction Problem (CSP) has one (or more) constraint function but no objective functions. Any solution that satisfies the constraints is equally good. (iii) A Free Optimisation Problem (FOP) has one (or more) objective functions but not constraint functions.

Table 4.1: Single-Objective Problem Taxonomy.

	Objective function	
Constraints	yes	no
yes	COP	CSP
no	FOP	-

Instead, with reference to the number of objective functions and the presence or not of a constraint function, problem classifications are in Table 4.2. (i) A CSOP is a constraint problem with a single objective function. (ii) A CMOP is a constraint problem with 2 or 3 objective functions. (iii) A Constraint Many Objective Problems (CMaOP) is a constraint problem with more than 3 objective functions. (iv) A SOP is an un-constraint problem with a single objective function. (v) A MOP is an un-constraint problem with 2 or 3 objective function. (vi) A MaOP is finally an un-constraint problem with more than 3 objective functions.

This rich classification framework of optimisation problem formulations allows us finally to abstract, model and solve many practical problems arising in engineering: Travelling Salesman Problem (TSP), Vehicle Routing Problem (VRP), Knapsack problem and all their variants for example in Operational Research (OR).

Table 4.2: Multi-Objective Optimisation Problem Taxonomy.

	N obj		
Constraints	1	2 or 3	4 or more
yes	CSOP	CMOP	CMaOP
no	SOP	MOP	MaOP

4.2 Algorithmic Solvers Classification

This section introduces to the structure and classification of optimisation solvers, focusing on global meta-heuristic methodologies and in particular on MA. For a comprehensive introduction to the field, please refer to [32].

Most of the optimisation algorithms, from an high level point of view, are structurally similar since they share the same components or operators. (i) The **Initialisation** block includes the procedures for the definition of the search space and of the first solution(s). (ii) The **Generation** block includes the methodologies for the generation of the new solution based on the previous one(s) or on the history. (iii) The **Evaluation** block quantifies the candidate solution(s). It is also responsible to take into consideration, if applicable, the handling of uncertainties, of multiple objectives and of constraints. (iv) The **Selection** block selects a subset of solution (if population-based) for the generation of the next candidate solutions. (v) The **Control** block, finally, includes the methodologies for the control and tuning of the algorithm parameters.

Optimisation solvers can then be classified based on many criteria. The ones presented in the previous section about the optimisation problem formulation are still applicable here for the algorithmic optimisation solvers. (i) Besides them, an algorithm can be **Deterministic or Stochastic**. The former group provides consistent and predictable results, while the latter introduces randomness or probabilistic elements, leading to potentially varied outcomes. (ii) An algorithm can also be **Exact, Heuristic or Meta-heuristic**. Exact algorithms are a special class of systematic or exhaustive optimisation techniques. They guarantee, for specific types of optimisation problems, to find the optimal solution with a predictable amount of computational resources. Heuristics are designed and tuned for specific problems and they are usually greedy. Meta-heuristics [36] are nature-inspired and are structured in such a way to be general enough to be applied to multiple problem instances.

The following of this section focuses on meta-heuristic algorithms which are the most widely used and promising. They represent also the building blocks of the methodologies presented in the following of the thesis. Extending [35], a possible classification of meta-heuristics approaches considers 5 main categories as in Fig. 4.1: (i) Hill-climbing, (ii) Trajectory, (iii) Population, (iv) Surrogate and (v) Hybrid. Category **(i)**, "Hill-climbing" is based on the analogy of "The Mountaineer": as a climbing mountaineer, it tests many possible future moves and choose only the most locally advantageous. It focuses the search strategy on greedy exploitation with minimal exploration permitting then a fast convergence to the local optimum. Category **(ii)**, "Trajectory", is based on the analogy of "The Sightseer": the sightseer is looking for the best place to visit

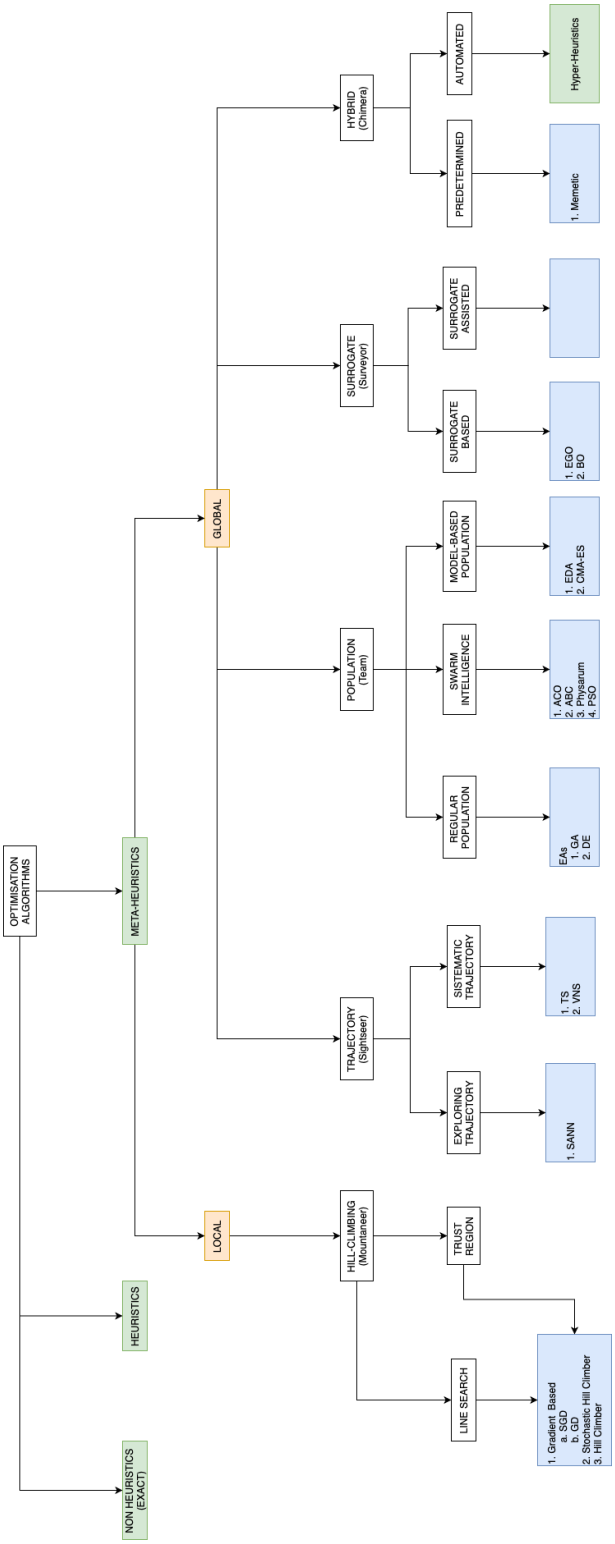


Figure 4.1: Taxonomy of Optimisation Algorithms

but in such a journey he is moving through many and diverse directions. This category can be further decomposed in: (a) **Exploring** and (b) **Systematic** Methods. Type (a) includes a combination of exploration and exploitation methods to allow for a global search. For example, Simulated Annealing (SA) is based on the analogy with the engineering process of the annealing. It is an extension of the basic *Hill Climbing* procedures where an *acceptance function* allows to choose a solution with a lower performance during the selection step. The probability of choosing a less promising solution is then tuned during the optimisation run time. Type (b) instead explores globally the search space by subdividing it in sub-areas. Examples are Tabu Search (TS) and Variable Neighbourhood Search (VNS). Category **(iii)**, "Population", can be understood with the analogy of "The Team" or the Darwin's evolutionary theory. As a working team, they are formed by a group of individuals which team-up to achieve their mutual goal together. They split up to explore different locations and share their knowledge with other members of the team. As a biological species evolving in its environment, also, the genetic information is passed through artificial generations eventually converging to the global optimum. This group can be further divided in three branches: (a) **Regular**, (b) **Swarm Intelligence** and (c) **Model-Based**. Type (a) is the closest to Darwin's theory. Important example are: Evolutionary Algorithms (EAs), Genetic Algorithm (GA) and Differential Evolution (DE). Type (b), Swarm Intelligence (SI), is based on the interaction of many agents with both individual and group behaviours which gives rise to an emergent intelligence of the swarm. Examples are: Ant Colony Optimisation (ACO), Artificial Bee Colony (ABC), Particle Swarm Optimisation (PSO) and Physarum. Type (c) uses instead a mathematical/statistical model to calculate the most promising regions where to sample for the new candidate solutions. Examples are: Estimation of distribution algorithm (EDA) and Covariance Matrix Adaption-Evolution Strategy (CMA-ES). Category **(iv)**, "Surrogate", is based on the analogy of "The Surveyor". The Surveyor is a specialist who systematically measures a landscape by taking samples of the height to create a topological map. This map resembles the real landscape with a given approximation accuracy and is typically exact at the sampled locations and models the remaining landscape by regression. It can then be examined and utilised to approximate the quality of an unknown point and further be updated if new information is acquired. Ultimately it can be used to guide an individual to the desired location. This category substitute or integrate the model of the problem under optimisation with a less computationally expensive surrogate model. This group can be further divided in (a) **Surrogate-Based** for which two examples are Efficient Global Optimisation (EGO) and Bayesian Optimisation (BO), and (b) **Surrogate-Assisted**. Category **(v)**, "Hybrid" is finally based on the analogy of "The Chimera". As the Chimera is a mixture, composition, or crossover of other individuals, this group of algorithms uses a combination of the previous methodologies as building blocks to generate a more complex algorithmic structure. It can be decomposed in two further types: (a) **Predetermined** and (b) **Automated**. Type (a) makes an a-priori combination of the different algorithmic parts. An important example is the group of MAs which combines EAs and local search and for which different approaches are defined as Darwinian, Baldwinian or Lamarchian. MAs are based on the concept of 'meme' which was introduced by Richard Dawkins in 'The Selfish Gene' [126]. Type (b)

instead automatically select and tune, through a list of alternatives, the best promising optimisation technique for each specific optimisation operator and part of the problem. An examples is given by the Hyper-Heuristics.

4.2.1 Memetic Approaches Based on Differential Evolution

This sub-section recalls the working principles of DE, a strategy introduced by Storn and Price in [127] that exhibits very good performance over a wide variety of optimisation problems [128]. It then introduces to MP-AIDEA a memetic hybridisation of DE with local search. An extension of MP-AIDEA will be presented and used in Chapter 5.

Following the notation introduced in [129], we can express the general DE process as a discrete dynamical system. The governing equation, for the i-th individual at generation k, is:

$$\mathbf{x}_{i,k+1} = \mathbf{x}_{i,k} + S\left(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}, \mathbf{x}_{i,k}\right) \mathbf{u}_{i,k} \tag{4.2}$$

with

$$\begin{aligned} \mathbf{u}_{i,k} = \mathbf{e}\,[&G\mathbf{x}_{r_1,k} + (1-G)\mathbf{x}_{i,k} + F\left(\mathbf{x}_{r_2,k} - \mathbf{x}_{r_3,k}\right) \\ &+(1-G)F\left(\mathbf{x}_{\text{best},k} - \mathbf{x}_{i,k}\right)] \end{aligned} \tag{4.3}$$

where G can be either 0 or 1 (with G=1 corresponding to the DE strategy DE/rand and G=0 corresponding to the DE strategy DE/current-to-best. In Eqs. (4.2) and (4.3), r_1, r_2 and r_3 are integer numbers randomly chosen in the population, and e is a mask containing random numbers of 0 and 1 according to:

$$e_t = \begin{cases} 1 \Rightarrow U \leq CR \\ 0 \Rightarrow U > CR \end{cases} \quad t = 1, \ldots, n_{\text{D}} \tag{4.4}$$

U is a random number taken from a random uniform distribution [0, 1]. The product between e and the term in square brackets in Eq. (4.3) has to be intended component-wise. In this work, given $u_{t,i,k}$, the t-th component of the trial vector $\mathbf{u}_{i,k}$, the following correction is applied to satisfy the box constraints:

$$u_{t,i,k} = \begin{cases} \left(x_{t,i,k} + x_{t,\text{ lower}}\right)/2, & \text{if } u_{t,i,k} < x_{t,\text{ lower}} \\ \left(x_{t,i,k} + x_{t,\text{ upper}}\right)/2, & \text{if } u_{t,i,k} > x_{t,\text{ upper}} \end{cases} \tag{4.5}$$

The selection function S is defined as:

$$S\left(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}, \mathbf{x}_{i,k}\right) = \begin{cases} 1 & \text{if } f\left(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}\right) < f\left(\mathbf{x}_{i,k}\right) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

In the general case in which the indices r_1, r_2 and r_3 can assume any value, in [129] it was demonstrated that the population can converge to a fixed point different from a local minimum or to a level set. Furthermore, in [130] it was demonstrated that DE can converge to a hyperplane that does not contain the global minimum.

As explained above, a MA is a particular case of Predetermined Hybrid algorithm which combine EA and local search. An example of MA with the use of DE is given by MP-AIDEA [131]. A first generalisation of DE to overcame its limitation was pro-

posed in [129] with Adaptive Inflationary Differential Evolution Algorithm (AIDEA) where DE and Monotonic Basin Hopping (MBH) [132] approaches are combined. The performance of AIDEA is however dependent upon the parameters controlling both the DE and MBH heuristics. These parameters are the crossover probability CR, the differential weight F, the radius of the local restart bubble δ_{local} and the number of local restarts n_{LR}, whose best settings are problem dependent. A further algorithmic evolution was presented in [131] with MP-AIDEA. In this paper, a simple mechanism is presented to adapt CR and F within a single population and a multi population strategy to adapt δ_{local} and n_{LR}. MP-AIDEA will be further developed in Chapter 5 and Appendix B in order to deal respectively with constraint handling and MOP.

4.3 Robust Optimisation

It is often very important to include a quantification of uncertainty within the problem formulation in Eq. (4.1). Then, if we define the uncertainty vector $\mathbf{u} \in U$, with U the uncertainty space, the CSOP can be translated in:

$$\begin{aligned} &\text{minimise}_{\mathbf{d}\in D}\ \phi(\mathbf{d},\mathbf{u}) \\ &\text{subject to}\quad \gamma_j(\mathbf{d},\mathbf{u}) \leq 0,\ \ j=1,...,r \\ &\qquad\qquad\ \ \eta_j(\mathbf{d},\mathbf{u}) = 0,\ \ j=1,...,p \\ &\qquad\qquad\ \ \mathbf{u} \in \mathbb{U} \end{aligned} \tag{4.7}$$

where ϕ, γ_j and η_j are some measures that account for the effect of $\mathbf{u}$ respectively on the quantities of interest f, c_j and g_j.

For Eq. (4.7) to be solved, two possible frameworks can be used: deterministic or probabilistic [11]. The best choice between probabilistic and deterministic approaches is case specific. However, as a general statement, the former could give good solution on average but lead to failures for some specific uncertain values while on the other hand, the latter is more conservative. The deterministic approach solves the problem by optimising for the worst possible condition. The set of possible scenarios can be defined explicitly using a discrete and then combinatorial formulation [133, 134] or implicitly where interval-valued variables can vary continuously within their lower and upper bounds. For the latter problem formulation [134] presents two important mathematical representations: *min-max* (presented in Section 4.4) and *min-max regret*. With the deterministic approach, functions ϕ, γ and η in Eq. (4.7) are considered at a particular value:

$$\begin{aligned} F_\phi(\mathbf{d}) =& \phi(\mathbf{d},\mathbf{u}^*) = \ \ sup_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}) \\ C_\gamma(\mathbf{d}) =& \gamma(\mathbf{d},\mathbf{u}^*) = \ \ sup_{\mathbf{u}\in U} c(\mathbf{d},\mathbf{u}) \\ G_\eta(\mathbf{d}) =& \eta(\mathbf{d},\mathbf{u}^*) = sup_{\mathbf{u}\in U} \|g(\mathbf{d},\mathbf{u})\| \end{aligned} \tag{4.8}$$

where $\mathbf{u}^*$ is the uncertain vector for which f, c and $\|g\|$ attain the maximum value. In the probabilistic problem there is enough information to model uncertainty with probability distributions. Two examples of probabilistic measures of robustness are:

1. The conditional expectation E of a utility function $\phi(f)$:

$$F_\phi(\mathbf{d}) = E[\phi(f)|\mathbf{d}] \tag{4.9}$$

where the value of the expectation is conditional to the choice of the design (or decision) vector $\mathbf{d}$. Different definitions of the function ϕ have been proposed in the literature, see [135] for some examples. When the utility function is simply f one can account for both the expected value of f and its variance σ with the two-objective problem:

$$\begin{array}{l} \min_{\mathbf{d}} E(f|\mathbf{d}) \\ \min_{\mathbf{d}} \sigma(f|\mathbf{d}) \end{array} \tag{4.10}$$

Methods for solving problem (4.10) can be found in [135–139].

2. The probabilistic threshold, where, for a given threshold q, the conditional probability that the function f assumes values lower than q is maximised:

$$\max_{\mathbf{d}} Pr(f < q|\mathbf{d}). \tag{4.11}$$

This approach can be easily extended by adding q as an objective function to be minimised.

Note that these two approaches are in fact equivalent if one takes the utility function $\phi = f < q$. In this case the utility function is the indicator function and the expectation of the indicator is the probability that $f < q$.

Robust Optimisation with Evidence Theory

This section presents an extension of the problem formulation for OUU with the probabilistic approach in Section 4.3 for conditions when there is not enough information to quantify uncertainty with a probability distribution. The probabilistic condition has then to be relaxed and IP theories used for UQ. We will focus here on the use of DST for OUU introduced by previous authors in the context of engineering applications [140]. The reader can finds a short introduction of DST in Chapter 3. This section will be the base for EBRO method presented in Part III.

The central idea is to maximise the belief in statement Eq. (3.13). This condition alone, however, is not enough to qualify the realisations of the performance indicator f. In fact, the condition $f \in \Phi$ alone would not say much on the optimality of the values of f.

Consider now the simple case in which $\Phi = \{f|f \leq \nu\}$ and $f : U \times D \to \mathbb{R}$ is a function of some decision vector $\mathbf{d} \in D \subseteq \mathbb{R}^{n_d}$ and some uncertain vector $\mathbf{u} \in U \subseteq \mathbb{R}^{n_u}$. If f is a performance index it is now easy to define the optimality condition:

$$\begin{array}{l} \min_{\nu \in \mathbb{R}} \nu \\ s.t. \\ f(\mathbf{d}, \mathbf{u}) \leq \nu \end{array} \tag{4.12}$$

that leads to the robust optimisation problem

$$\begin{aligned} &\max_{\mathbf{d}\in D} Bel(f(\mathbf{d},\mathbf{u}) \leq \nu) \\ &\min_{\nu\in\mathbb{R}} \nu \end{aligned} \tag{4.13}$$

Problem (4.13) can be extended to include constraints in three different forms:

$$\begin{aligned} &\max_{\mathbf{d}\in D} Bel(f(\mathbf{d},\mathbf{u}) \leq \nu) \\ &\min_{\nu\in\mathbb{R}} \nu \\ &s.t. \\ &c(\mathbf{d},\mathbf{u}) \leq \nu_c \end{aligned} \tag{4.14}$$

$$\begin{aligned} &\max_{\mathbf{d}\in D} Bel(f(\mathbf{d},\mathbf{u}) \leq \nu) \\ &\min_{\nu\in\mathbb{R}} \nu \\ &s.t. \\ &Bel(c(\mathbf{d},\mathbf{u}) \leq \nu_c) > 1-\epsilon \end{aligned} \tag{4.15}$$

$$\begin{aligned} &\max_{\mathbf{d}\in D} Bel(f(\mathbf{d},\mathbf{u}) \leq \nu_f) \\ &\max_{\mathbf{d}\in D} Bel(c(\mathbf{d},\mathbf{u}) \leq \nu_c) \\ &\min_{\nu_f\in\mathbb{R}} \nu_f \\ &\min_{\nu_c\in\mathbb{R}^{n_c}} \nu_c \end{aligned} \tag{4.16}$$

Eq. (4.14) introduces the deterministic constraint vector function C, Eq. (4.15) introduces a set of constraints on the belief that the constraints are satisfied, while Eq. (4.16) tries to maximise the belief that the constraints are satisfied. Note that Eq. (4.15) might not have any solution even if Eq. (4.14) has a solution because the constraint on the belief of the satisfaction of the constraints implies that constraints need to be satisfied for a set of values and not for a single one. Because of condition in Eq. (3.11) it is clear that one can derive an equivalent formulation with *Plausibility*.

Optimisation Algorithms with Evidence Theory

This section is dedicated to the algorithmic solvers for OUU for UBD with the use of DST. DUU makes designers able to handle a higher degree of complexity but, on the other hand, it is particularly challenging due to its high computational cost. Particularly important is the effort to include the different types of uncertainty, aleatory and epistemic, in the design process. Also restricting our attention to the use of only epistemic uncertainty, the task is challenging and at the moment not completely solved. Recent examples of the application of system-level optimisation principles, including uncertainty, to the design of space systems, can be found in [141] and [142]. Note, however, that the former proposes an exponentially complex computational method that cannot be used for large-scale systems and the latter does not include epistemic uncertainty. Approaches specifically for UQ with DST have been presented in Section 3.3. Here the problem is expanded to solve the UQ within the optimisation framework. In the literature on Reliability-Based Optimisation some authors proposed methods to efficiently solve the constraint in Eq. (4.15) by introducing hypotheses on the local differentiability of the constraint functions, the existence of a Most Probable Focal Element

(MPFE) or by a form of probabilistic approximation of the belief functions [143–145] to speed up the calculation of an approximation of Bel. In [77] the authors present three approaches to solve problem in Eq. (4.13):

- The *direct approach* uses a multi-objective optimiser to find the trade-off between the threshold ν and corresponding $Bel(f < \nu)$ where the standard dominance index is defined as:

$$I_i = |\{j | Bel(\mathbf{d}_j) > Bel(\mathbf{d}_i) \wedge \nu_j < \nu_i,\ j = 1, ..., n_{pop} \wedge j \neq i\}| \tag{4.17}$$

  with $|.|$ the cardinality and n_{pop} the number of design vectors. But this approach has two main problems: each design vector in (4.17) is related to a Belief - ν curve, and different design vectors could give the same Pareto front.

- The *step method* reduces the computational effort solving a single objective problem: an initial $\mathbf{d}$ is chosen that corresponds to a threshold ν_1 with $Bel = 1$ and then the threshold is reduced step by step running, for each new ν_k, local optimisation and maximising the corresponding Belief. The new optimisation is started from the previous optimal $\mathbf{d}$ configuration and a local optimiser is used; this reduces the possibility to evaluate the real global optimum, but it is on the other hand a necessary simplification to avoid the explosion in computational time.

- The *cluster approximation*, finally, looks at the whole search space (design and uncertainty) and for different thresholds ν_i clusters all the possible sets, in $\mathbf{D} \times \mathbf{U}$, that satisfy the condition: $f < \nu_i$. For each ν_i and design, then, the belief can be easily evaluated by adding the FEs included in the cluster and finally, the $\mathbf{d}$ that maximise the belief approximation is chosen.

4.4 Min-Max

A min-max optimisation aims at minimising, with respect to a vector $\mathbf{d}$ defined in some space D, the maximum value of a given cost function with respect to a vector $\mathbf{u}$, different from $\mathbf{d}$. In its constraint version, the constraints have to be always satisfied for all values of $\mathbf{u}$ in a given set U. More formally, the multi-objective constrained min-max problem can be formulated mathematically as:

$$\begin{aligned} &\min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f_i(\mathbf{d}, \mathbf{u}) && \forall i \in I_f = [1, ..., m]^T \\ &s.t. \\ &c_i(\mathbf{d}, \mathbf{u}) \leq 0 && \forall \mathbf{u} \in U, \forall i \in I_c = [1, ..., s]^T \end{aligned} \tag{4.18}$$

where f_i is the i-th objective function and c_i is the i-th constraint function. Both f_i and c_i are defined on the space $D \times U$ and depend on a vector of design (or decision) variables $\mathbf{d} \in D \subset \mathcal{R}^n$ and a vector of uncertain (or environmental) variables $\mathbf{u} \in U \subset \mathcal{R}^m$. The solution $\mathbf{d}_{\text{opt}}$ of Eq. (4.18) has two properties: it satisfies all the constraint functions c_i over the whole uncertain domain U and minimises the worst realisation of the objective function f over U.

This type of problem arises in various fields, such as game theory, economics, engineering, and computer science. For example, it has been used for the study and solution of famous board games such as the Egyptian ancient game Seega [146], the game of checkers [147] and chess [148]. It is also used for the design of the robustness of engineered systems affected by uncertainty: design of electric circuits [149], design of online controllers [150], of aerodynamic shapes [151], of the location of sensors [152] and pursuit-evasion games between missile and target [153].

The min-max optimisation problem can be interpreted in different ways:

- a bi-level problem as described in Section 4.1 where two agents quantify their utility with opposite fitness functions.
- a single-level problem where the feasible domain due to constraint satisfaction is the result of a further optimisation problem.
- a limit condition of the OUU where the most conservative approach is applied. This approach is called *robust regularisation* in [154] where the function ϕ is defined as the worst case scenario in a neighbourhood $U(\epsilon)$ where ϵ is called the *regularisation parameter*. This approach can also be found in [114, 155, 156] and its generalisation brings to the global worst-case scenario.
- a two-player zero-sum game within the game theory framework where the optimal strategy brings the two agents to a Nash equilibrium.
- a deterministic approach to decisions making under uncertainty. In this case, which is connected to the previous two points, one can understand the engineering problem as a zero-sum game where the two antagonist players are respectively the designer, handling the decision variables, and Nature, handling the uncertainty variables.

Optimisation Algorithms

Many algorithmic paradigms can be found in the literature to solve the min-max problem. One example is given by mathematical programming methods [149] for which an overview can be found in [134]. The use of Mathematical Programming, however, requires strong assumptions on the nature of the problem and tends to be problem specific. On the other hand meta-heuristic approaches [157–159] and in particular evolutionary based algorithms [42, 152, 157, 158, 160] and co-evolutionary approaches [133] appear to be more flexible. For example one can find a form of Genetic Algorithm (GA) in [157], a special version of the DE called Crowding Differential Evolution (CrDE) combined with the Nash ascendancy relation in [158] and a form of Particle Swarm Optimisation (PSO) in [160]. As stated in [42] we can divide the class of evolutionary approaches for min-max problems in three branches. The first one includes problems with a discrete set of possible scenarios U. In [160] U is by definition discrete and small, in [152], initially continuous, it is instead discretised by a uniform grid while in [157] it is reduced to be discrete by a random selection. The second branch considers a continuous space U and directly solves a sequence of nested problems [149, 150]. This

approach could be computationally intractable if no strategy is implemented to alleviate the cost. One interesting procedure for decoupling the min-max problem formulation in Eq. (4.18) has been suggested by Shimizu and Aiyoshi [41] and further elaborated in [42, 43]. The idea is to alternate a *minimisation* and a *restoration* (or maximisation) process. As the alternation of these two processes progresses, we incrementally build a discrete representation of the space of the maxima in U. This representation is then used in the minimisation process to converge to an approximation of the desired min-max solution. This mechanism takes into memory the previous solutions avoiding the optimiser to follow the same path again and again. For this reason, the proposed algorithm does not suffer from the Red Queen Effect. Finally, a third category uses a form of co-evolution optimisation [153, 157, 161, 162] where two separate populations evolve simultaneously with a predator-prey interaction in D and U separately each one being a competitor and environment to the other. As demonstrated in [163] however most of the co-evolutionary algorithms for min-max have to satisfy a symmetric assumption known as Issac's condition [164]. A class of problems that can be properly modelled under this assumption exists [161, 165] and for them, the co-evolution approach is proved to converge to the global solution [166]. However, this is not the general case and new methods to overcome this limitation are presented in [167]. In the interest of completeness, we include also the *Best Replay* approach cited by [168] where two optimisation problems are alternated until convergence, one minimising over D and the other maximising over U. It has been proven, however, that this approach often does not converge or it cycles through wrong design candidates, a problem known as the Red Queen Effect.

Another research line, applicable to any previously presented method, deals with the reduction of computational complexity. It makes use of surrogates which approximate the fitness function [42, 43, 158]. In particular, [42, 43] consider response surfaces within an EGO framework where they alternate the minimisation in D of the surrogate of the maxima in U and the maximisation in U of the real function for the obtained optimal solution in D.

When it comes to constraint handling, in the existing literature, only a few papers could be found that have explored how to deal with constraints in min-max optimisation. Most of them need to start from some strong assumptions on the nature of constraints and cost functions and have been developed for constrained bi-level problems and not specifically for the treatment of min-max problems [169, 170]. In [153] the duality between the primal constrained min-max problem and dual unconstrained min-max problem with the Lagrange multipliers is demonstrated for both separable and non-separable constraints under Issac's condition. The constrained problem is then translated in the unconstrained one and a co-evolutionary algorithm is finally used to converge to the saddle point. Interestingly, this link is bi-directional: other works indeed translate a single-level constrained problem in an unconstrained min-max problem and use the co-evolution approach to solve it [161, 162].

4.5 Multi Disciplinary Optimisation

Often engineers have to address the optimisation of complex systems consisting of multiple interconnected components or disciplines. Two optimisation methodologies that have been developed for this porpoise are SDO and MDO. The former is applied to problems with many disciplines weakly coupled while the latter on the other side is used to problems with fewer disciplines but more strongly connected. Both these approaches recognise that the performance of such systems depends not only on the individual disciplines but also on their interactions.

The origins of MDO can be traced back to the work of Schmit, Haftka, and their collaborators, who expanded structural optimisation to include other disciplines. Early applications were in aircraft wing design, which involves the interplay of aerodynamics, structures, and controls. Since then, MDO and SDO have been applied to various engineering systems, such as complete aircraft, bridges, buildings, railway cars, microscopes, automobiles, ships, propellers, rotor-craft, wind turbines, and spacecraft to name a few.

MDO and SDO problems are usually modelled with the following well-defined mathematical formulation [40]:

$$\begin{array}{ll} \min & f_0(\mathbf{x},\mathbf{y}) + \sum_{i=1}^{N} f_i\left(\mathbf{x}_0,\mathbf{x}_i,\mathbf{y}_i\right) \\ \text{w.r.t.} & \mathbf{x},\hat{\mathbf{y}},\mathbf{y},\overline{\mathbf{y}} \\ \text{s.t.} & \mathbf{c}_0(\mathbf{x},\mathbf{y}) \geq 0 \\ & \mathbf{c}_i\left(\mathbf{x}_0,\mathbf{x}_i,\mathbf{y}_i\right) \geq 0 \quad \text{for i} = 1,\ldots,\text{N} \\ & \mathbf{c}_i^c = \hat{\mathbf{y}}_i - \mathbf{y}_i = 0 \quad \text{for i} = 1,\ldots,\text{N} \\ & \mathbf{R}_i\left(\mathbf{x}_0,\mathbf{x}_i,\hat{\mathbf{y}}_{j\neq i},\overline{\mathbf{y}}_i,\mathbf{y}_i\right) = 0 \quad \text{for i} = 1,\ldots,\text{N} \end{array} \tag{4.19}$$

which is known as the All-At-Once (AAO) problem. In Eq. (4.19), N is the number of disciplines, $\mathbf{x}_i$ are the discipline variables, $\mathbf{x}_0$ are variables shared by more than one discipline, $\mathbf{y}_i$ are the coupling variables (output from a single discipline analysis), $\overline{\mathbf{y}}_i$ are the state variables (used only inside one discipline analysis), $\mathbf{x}$ is the concatenation of all the discipline variables, $\mathbf{x} = \left[\mathbf{x}_0^T,\mathbf{x}_1^T,\ldots,\mathbf{x}_N^T\right]^T$, $\mathbf{y}$ is the concatenation of all the coupling variables, $\mathbf{y} = \left[\mathbf{y}_0^T,\mathbf{y}_1^T,\ldots,\mathbf{y}_N^T\right]^T$, f_0 is the global objective function, $\mathbf{c}_0$ are the global constraints, f_i are the discipline objectives, $\mathbf{c}_i$ are the discipline constraints, $\mathbf{c}_i^c$ are the consistency constraints, and $\mathcal{R}_i$ are the discipline analysis constraints. This form of the design optimization problem includes all coupling variables, coupling variable copies, state variables, consistency constraints, and residuals of the governing equations directly in the problem statement.

As presented in [40], the usual approach for the solution of the AAO in Eq. (4.19) is by means of a suited problem decomposition. The paper lists the most common decomposition problem formulations.

Problem Eq. (4.19) can be generalised by introducing the uncertainty handling:

$$\begin{array}{ll} \min & \Xi_0\left[f_0(\mathbf{x},\mathbf{u},\mathbf{y})\right] + \sum_{i=1}^{N} \Xi_i\left[f_i\left(\mathbf{x}_0,\mathbf{x}_i,\mathbf{u}_0,\mathbf{u}_i,\mathbf{y}_i\right)\right] \\ \text{w.r.t.} & \mathbf{x},\hat{\mathbf{y}},\mathbf{y},\overline{\mathbf{y}} \\ \text{s.t.} & \Lambda_{c,0}\left[\mathbf{c}_0(\mathbf{x},\mathbf{u},\mathbf{y}) \geq 0\right] - \Lambda_{Reqc,0} \geq 0 \\ & \Lambda_{c,i}\left[\mathbf{c}_i\left(\mathbf{x}_0,\mathbf{x}_i,\mathbf{u}_0,\mathbf{u}_i,\mathbf{y}_i\right) \geq 0\right] - \Lambda_{Rec,i} \geq 0 \quad \text{for i} = 1,\ldots,\text{N} \\ & \Lambda_{cc,i}\left[\hat{\mathbf{y}}_i(\mathbf{u}) - \mathbf{y}_i(\mathbf{u}) = 0\right] - \Lambda_{Reqcc,i} \geq 0 \quad \text{for i} = 1,\ldots,\text{N} \\ & \Lambda_{R,i}\left[\mathcal{R}_i\left(\mathbf{x}_0,\mathbf{x}_i,\mathbf{u}_0,\mathbf{u}_i,\hat{\mathbf{y}}_{j\neq i},\mathbf{y}_i,\overline{\mathbf{y}}_i\right) = 0 - \Lambda_{ReqR,i} \geq 0 \quad \text{for i} = 1,\ldots,\text{N} \end{array} \tag{4.20}$$

where in addition to what was described before, Ξ_0 and Ξ_i are operators over the uncertainty variables, $\mathbf{u}_0$ are the components of the uncertain vector shared by more than one discipline and $\mathbf{u}_i$ are the discipline's uncertain variables.

Optimisation Algorithms

Implementing MDO requires careful organisation of discipline-analysis models, approximation models, and optimisation software to achieve an optimal design, which is called MDO architecture. The architecture can be either **Monolithic** or **Distributed**. In a monolithic approach, a single optimisation problem is solved, while in a distributed approach, the problem is divided into multiple sub-problems. A review of the most common architectures is in [40].

Recent examples of the application of system-level optimisation principles, including uncertainty, to the design of space systems, can be found in [79, 141, 142, 145, 171, 172]. [145] makes the assumptions of normally distributed epistemic parameters. [141] is focused on RBDO and suggests a sequential method that incorporates mixed aleatory-epistemic uncertainty. The computational cost remains however exponentially complex with the problem dimension. [142] does not include epistemic uncertainty. [171] also could be intractable with high-order problems.

4.6 Multi Objective Optimisation

There are mainly three approaches for Multi-Objective Optimisation (MOO) [173, 174].

- The *a posteriori* methods, based on the definition of a partial order, calculate a set of equally valuable solutions. The decision maker then, informed of this trade-off, chooses within the set. In the posterior approaches the whole set of possible solutions can be generated by two algorithmic methods: the direct multi-objective approach [175, 176] or the parameter-based scalarisation procedure. By scalarisation we mean that the different objectives are aggregated and then a SOP is solved. By using different parameters of the aggregation function finally the MOP is translated to a number of SOPs and the set of optimal solutions is reconstructed [177].

- In the *a priori* methods the decision maker is required to specify additional preferences to define a total order between different options, for example by defining an utility function. The optimisation eventually finds a single minimal solution.

- The *interactive* methods finally require feedback and preferences from the user multiple time during the execution of the algorithm.

Finally, the assessment of the quality of a MOO algorithm is a delicate matter. Useful indications on how to categorise difficulties in MOPs have been described in [178]. A benchmark based on these information has been defined in [179] while the complexity introduced by a constrained search space has been included in [180].

4.6.1 Normalisation

In case of prior knowledge about the reference points $\mathbf{z}^*$ (best) and $\mathbf{z}^{**}$ (worst), the objective functions $\mathbf{f}$ can be normalised in order to reduce the difference in the order of magnitude between the components f_i:

$$\bar{\mathbf{f}} = \frac{\mathbf{f} - \mathbf{z}^*}{\mathbf{z}^{**} - \mathbf{z}^*}. \tag{4.21}$$

$\mathbf{z}^*$ and $\mathbf{z}^{**}$ can be defined as reference solutions by the decision maker. However $\mathbf{z}^*$ usually corresponds to the ideal point $\mathbf{z}_{ideal}$ or to the utopian point $\mathbf{z}_{\text{utopian}}$ while $\mathbf{z}^{**}$ corresponds to the nadir point $\mathbf{z}_{nadir}$.

4.6.2 Scalarisation Strategies

This section reviews the most important parameter-based scalarisation approaches: Epsilon-Constraint Scalarisation (ECS), Weighted-Sum Scalarisation (WSS), Benson Scalarisation (BS), Weighted Chebichev Scalarisation (WCS) and Pascoletti-Serafini Scalarisation (PSS).

We consider a generic preference vector $\boldsymbol{\omega} = [\omega_1, ..., \omega_m]^T$ for the objective functions $\mathbf{f} = [f_1, f_2, ..., f_m]^T$ and a generic reference point $a = \mathbf{z}^*$. The values of $\boldsymbol{\omega}$ and a can be either defined as a priory by the decision maker or made varying in order to reconstruct the entire efficient set.

The scalarisation methods are compared in Table 4.3 as in [181] where the following criteria have been considered: the possibility to use different ordering cones, the necessity or not of boundedness and convexity conditions, the provability for obtaining properly efficient solutions, the use of reference and preference information and the introduction by the method of additional constraint functions.

Epsilon Constraint Scalarisation

The ECS was introduced by Haimes et al. in 1971 [182]. In this approach, one of the functions in $\mathbf{f}$ in Eq. (4.1) is maintained as the objective while the remaining functions are treated as inequality constraints

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} f_i \\ &s.t. \quad f_k \leq \epsilon_k \quad k \in \{1, ..., m\} \setminus \{i\} \\ &\qquad c_j \leq 0 \quad \forall j \in \{1, ..., n\} \end{aligned} \tag{4.22}$$

Table 4.3: Characteristics of six scalarisation methods

Method	WSS	ECS	BS	WCS	PSS	CS
Ordering cone	any	$\mathbb{R}^m_+$	$\mathbb{R}^m_+$	$\mathbb{R}^m_+$	any	any
Boundedness from below	−	−	−	+	−	−
Convexity	+	−	−	−	−	−
Proof of properly efficient solutions	+	−	−	−	−	+
Preference weights	+	−	−	+	−	+
Reference points	−	−	−	−	+	+
Additional constraints or variables	−	+	+	+	+	−

The boundedness from below for the ECS is not an essential condition. However, the set of thresholds ϵ_k has to be decided carefully by the decision maker. A wrong selection, indeed, could bring to a not finite optimal solution or to an infeasible solution. The ECS can be applied only in the case when the ordering cone equals $\mathbb{R}^m_+$. The method does not require convexity condition on the problem under consideration. It generates weakly efficient solutions and does not provide conditions for generating properly efficient solutions. Decision makers preferences, namely weights of objectives and reference points, are not taken into account. Finally, the problem size increases due to adding the constraints.

Weighted-Sum Scalarisation

The WSS was suggested by Gass and Saaty [183] in 1955 and it is probably the most commonly used scalarization technique for MOP. Here the Eq. (4.1) translates to:

$$\min_{\mathbf{x}\in\mathbb{X}} \sum_{i=1}^{n} \omega_i f_i \tag{4.23}$$

As for the ECS the boundedness below is not required but in that case the weights $\boldsymbol{\omega}$ have to be chosen carefully. Weakly and properly efficient solutions are guaranteed under the convexity condition. Weights of objectives are used but reference points are not considered. The method does not introduce additional constraints.

Benson's Scalarisation

The method was introduced in [184]. Here an initial guess $\mathbf{x}_0$ is given by the decision maker. The sum of the deviations l_i is maximised to find a new dominating point:

$$\begin{aligned} &\max_{\mathbf{x}\in X} \textstyle\sum_{i=1}^{n} l_i \\ &s.t. \quad f_i(\mathbf{x}_0) - l_i - f_i(\mathbf{x}) = 0 \quad i = 1, ..., m \\ &\qquad l \geq 0 \\ &\qquad c_j \leq 0 \quad \forall j = 1, 2, ..., n \end{aligned} \tag{4.24}$$

The BS requires the ordering cone $\mathbb{K}$ to equal $\mathbb{R}^m_+$. The boundedness below is not a requirement, however if the condition is not satisfied, more attention has to be

put on the selection of $\mathbf{x}_0$. There is no necessity for the problem to be convex. BS provides necessary and sufficient conditions to converge to efficient solutions, but not to properly efficient solutions. Preferences from the decision maker are not taken into account. Finally, besides functions c_j, additional constraints are considered.

Weighted Chebyshev Scalarisation

The idea of the WCS is first presented in [185]. The Eq. (4.1) translates to:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} \|\mathbf{f}-\mathbf{z}_{\text{ideal}}\|_{\infty}^{\omega} \\ &s.t. \quad c_j \leq 0 \qquad \forall j=1,2,...,n \end{aligned} \tag{4.25}$$

where $\|\mathbf{f}-\mathbf{z}_{ideal}\|_{\infty}^{\omega}$ is the weighted Chebyshev distance $\max_i\{\omega_i(f_i-z_{ideal,i})\}$ between $\mathbf{f}(\mathbf{x})\in\mathbb{Y}$ and the ideal point $\mathbf{z}_{\text{ideal}}$.

The linearisation is often considered:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X},t\in\mathbb{R}} t \\ &s.t. \qquad \omega_i(f_i-z_{\text{ideal},i})\leq t, \quad \forall i=1,2,...,m \\ &\qquad\quad c_j\leq 0, \qquad\qquad\quad \forall j=1,2,...,n \end{aligned} \tag{4.26}$$

The WCS requires the cone $\mathbb{K}$ to be $\mathbb{R}_{+}^{m}$. The bondedness below is a necessary condition for the existence of $\mathbf{z}_{\text{ideal}}$. Instead the convexity assumption is not needed. The method assures generation of weakly efficient solutions and efficient solutions. However it is not guaranteed to generate properly efficient solutions. The preference vector $\boldsymbol{\omega}$ over the objective space is considered. The ideal point could be considered as a special case for the reference point. However the solutions are not guaranteed to be close to the reference point. In the linearised version, the size of the problem is increased by new constraints.

Pascoletti-Serafini Scalarisation

A first description of the PSS is given by Gerstewitz in [186]. As stated in [187], the PSS is a generalisation of ECS, WSS and WCS and it can be represented as:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} t \\ &s.t. \quad a+tr-f(x)\in\mathbb{K} \\ &\qquad c_j\leq 0, \qquad\qquad \forall j=1,2,...,n \end{aligned} \tag{4.27}$$

Eq. (4.27) can be interpreted as the process where the ordering cone $\mathbb{K}$ is moved in the direction $-r$ along the line $a+tr$ minimising the intersection $(a+tr-\mathbb{K})\cap f(\mathbb{X})$ until it becomes the empty set.

An arbitrary ordering cone can be adopted. The boundedness below and the convexity are not required conditions. The method guarantees to get at least weakly efficient solutions but it does not provide conditions to generate properly efficient solutions. It does use reference points but not preference vectors. Finally it uses additional functional constraints.

Conic Scalarisation

The Conic Scalarisation (CS) method was first introduced by Gasimov in [188] where beside the preference weighted vector $\boldsymbol{\omega}$ and the reference point $\mathbf{a}$, the augmentation parameter $\boldsymbol{\alpha}$ is considered:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X}} \textstyle\sum_i \omega_i(f_i - a_i) + \alpha \sum_i |\omega_i(f_i - a_i)| \\ &s.t. \quad c_j \leq 0 \qquad\qquad\qquad\qquad \forall j = 1, 2, ..., n \end{aligned} \tag{4.28}$$

As stated in [181], CS is a generalisation of WSS, BS and PSS. An arbitrary ordering cone can be used. The boundedness below is not an essential condition. No convexity is required. There are also conditions that guarantee to generate properly efficient minimal points. Preference and reference information is used. Finally, no additional constraints are required.

4.7 Constraint Handling

Many of the engineering problems we want to solve have restrictions and requirements that have to be satisfied which can be translated into constraints in the optimisation formulation. The goal of the solver is then to separate the space of potential solutions $\mathcal{S}$ into two or more disjoint regions, the feasible region (or regions) $\mathcal{F}$ containing those candidate solutions that satisfy the given constraints, and the infeasible region $\mathcal{U}$ containing those that do not satisfies the constraints.

Many and different methods to deal with constraints have been developed depending on the class of the optimisation solver, the nature of the optimisation variables and the constraint itself. [37] presents an overview of constraint handling for population based algorithms. Very specialised methodologies can be developed for each particular algorithm type. In [38], for example, the constraint handling in particle swarm algorithms is considered. [39] goes deeper on the penalty function approach. As stated in [36], constraint handling approaches can be categorised based on two main criteria: (i) Direct or Indirect and (ii) where they are applied. Regarding (i), in the case of indirect constraint handling, constraints are transformed into optimisation objectives. After the transformation, they eectively disappear, and all we need to care about is optimising the resulting objective function. This type of constraint handling is done before the EA run. In direct constraint handling, the problem oered to the EA to solve has constraints (is a COP) that are enforced explicitly during the EA run. Regarding (ii), the constraint handling procedure can be applied on the Genotype, on the Phenotype, on the Map from Genotype to Phenotype or on the Fitness Function.

In [36] there is also a review of the possible approaches to COP in the EA framework. The methods that are compared are: (i) Penalty Functions, (ii) Repair Functions, (iii) Restricting Search to the Feasible Region (iv) Decoder Functions.

For what concerns type (i), Penalty Functions, as Eq. (4.29) shows, they modify the fitness function f with a penalty P:

$$P(\bar{x}) = \begin{cases} 0 & \text{if} \quad e(\bar{x}) \leq 0 \\ w \cdot e(\bar{x}) & \text{if} \quad e(\bar{x}) > 0. \end{cases} \tag{4.29}$$

where w is a pre-defined or adapted parameters and e is a measure of constraint violation. The parameter w is particularly important because a proper setting should allow for a trade-off between exploration of the unfeasible region and not wasting time. A wrong setting of the parameter would lead to one of the two extremes causing suboptimal performance of the optimisation solver. We can classify between exterior and interior penalty function. The former is applied only when the candidate solution belongs to $\mathcal{U}$ and no modification is applied instead when it is included in $\mathcal{F}$. The interior penalty function instead modifies the fitness function f also in the case it belongs to $\mathcal{F}$. This is done to push the search in a suitable neighbour of the border between feasible and unfeasible regions. An other important distinction is between static, dynamic and adaptive penalty function approaches. Method (ii), Repair Functions, can be seen as a case where a local search is added to the main optimisation process. Based on the so-called Baldwinian versus Lamarkian learning approach discussed above, the mapping between Genotype and Phenotype is modified differently. In the Baldwinian case, the tness of the repaired solution is allocated to the infeasible point, which is kept, whereas with Lamarckian learning, the infeasible solution is overwritten with the new feasible point. Method (iii), Restricting Search to the Feasible Region is instead applied on Genotype only. Method (iv), Decoder Functions, finally, replace the original mapping from Genotype to Phenotype so that all solutions (i.e., phenotypes) are guaranteed to be feasible.

Part II

Global Optimisation Solver

5

Global Solution of Constrained Min-Max Optimisation

" Now, here, you see, it takes all the running you can do, to keep in the same place, if you want to get somewhere else,
you must run at least twice as fast as that! "

– Lewis Carroll, Through the Looking-Glass

The content of this chapter was published in:

- **G. Filippi** and M. Vasile, A Memetic Approach to the Solution of Constrained Min-Max Problems, in *2019 IEEE Congress on Evolutionary Computation, CEC 2019* - Proceedings, (Wellington), pp. 506513, 2019 [67].
- **G. Filippi** and M. Vasile, Global Solution of Constrained Min-Max Problems with Inflationary Differential Evolution, in *Optimisation in Space Engineering OSE (E. Minisci, A. Riccardi, and M. Vasile, eds.), no. Optimization and Engineering, Springer*, 2020, https://doi.org/10.1007/s11081-021-09613-3 [55].
- **G. Filippi** and M. Vasile, Inflationary Differential Evolution for Constrained Multi-Objective Optimisation Problems. BIOMA 2020 [55]

The min-max (or minimax) problem is an optimisation problem with the goal to minimise the maximum possible loss or regret. In other words, in a min-max problem the objective is to find the best decision or strategy that minimises the maximum possible loss that could occur under any circumstance or scenario.

This chapter proposes a novel algorithm for the solution of the class of CSOP min-max. Appendix B further extends the methodology for the solution of CMOP. The proposed algorithm is inspired by the procedure suggested by Shimizu and Aiyoshi [41] and further elaborated by Zhou and Zhang in [42] and by Marzat et al. in [43]. The idea is to alternate a *minimisation* and a *restoration* (or maximisation) process. As the alternation of these two processes progresses, we incrementally build a discrete representation of the space of the maxima in U. This representation is then used in the

minimisation process to converge to an approximation of the desired min-max solution. This mechanism takes memory of the previous solutions avoiding the optimiser to follow the same path again and again. For this reason, the proposed algorithm does not suffer from the Red Queen Effect. Differently from [42, 43] the method here presented intentionally avoids the use of surrogates for the reduction of the computational cost. The two approaches are however compatible and could be combined. Furthermore, a procedure for constraint relaxation for possible trade-offs between objective and constraint functions is here proposed. This chapter also presents an extension of MP-AIDEA [131] which is an hybrid memetic algorithm that combines DE and hill climbing methodologies with automatic control of algorithm' parameters. Two extensions are proposed in this context. The main contribution is the extension of the MP-AIDEA approach for handling constraints for which we propose here an indirect approach with an adaptive exterior penalty function for hard constraint handling. A further, preliminary, result is the adaptation of MP-AIDEA for the solution of MOP for which we propose a combination of WCS and PSS methods (in Appendix B). Finally, a benchmark for the testing of the method is developed and used.

Summarising, in this chapter we provide six main contributions to the min-max problem solution: i) a detailed algorithmic presentation of a memetic approach, ii) a constraint handling procedure, iii) a constrained relaxation strategy, iv) a scalarisation strategy for MOP (Appendix B), v) an algorithmic complexity analysis and vi) an extensive benchmark for SOP min-max on which our proposed solver has been tested. Points i), ii) and iv) apply for both the optimisation solver and the min-max procedure.

The reader can find the necessary background about the definition of the Optimisation problems and the algorithmic solvers in Chapter 4. In particular Section 4.4 is specifically dedicated to the min-max problem solution. It is also explained the current state of the art about MOP, MOO and constraint handling.

The remaining of this section presents the structure of the chapter. Section 5.1 introduces the formulation of the constrained min-max problem of interest. Section 5.2 then presents the proposed algorithm for SOP. Section 5.3 presents a complexity analysis. Section 5.4 then introduces a benchmark of synthetic functions with increasing complexity. Section 5.5 describes the results of the simulation on the proposed benchmark. The proposed method is tested on a realistic application case of space systems engineering under epistemic uncertainty in Section 5.6. Finally, conclusions are in Section 5.7. Appendix A presents all the tabulated results of the tests. Appendix B generalises to MOP by means of scalarisation approach.

5.1 Problem Definition

A min-max optimisation problem aims at minimising, with respect to a vector $\mathbf{d}$ defined in some space D, the maximum value of a given cost function $\mathbf{f}$ with respect to a vector $\mathbf{u}$, different from $\mathbf{d}$. In particular, we are interested in a class of constrained min-max problems where the constraints have to be always satisfied for all values of $\mathbf{u}$ in a given set U. More formally, the CMOP min-max can be formulated mathematically as:

$$\begin{aligned} &\min_{\mathbf{d}\in D}\max_{\mathbf{u}\in U} f_i(\mathbf{d},\mathbf{u}) && \forall i \in I_f = [1,...,m]^T \\ &s.t. \\ &c_i(\mathbf{d},\mathbf{u}) \leq 0 && \forall \mathbf{u} \in U, \forall i \in I_c = [1,...,s]^T \end{aligned} \tag{5.1}$$

where f_i is the i-th objective function and c_i is the i-th constraint function. Both f_i and c_i are defined on the space $D \times U$ and depend on a vector of design (or decision) variables $\mathbf{d} \in D \subset \mathcal{R}^n$ and a vector of uncertain (or environmental) variables $\mathbf{u} \in U \subset \mathcal{R}^m$. The solution $\mathbf{d}_{\text{opt}}$ of Eq. (5.1) has two properties: it satisfies all the constraint functions c_i over the whole uncertain domain U and minimises the worst realisation of the objective function f over U. Furthermore, we assume that both f_i and c_i are locally $\mathcal{C}^2$. This assumption can be relaxed if the local search can handle non-differentiable or discontinuous problems.

This chapter presents the methodology for the solution of the CSOP min-max for which Eq. (5.1) can be translated to:

$$\begin{aligned} &\min_{\mathbf{d}\in D}\max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}) \\ &s.t. \\ &c_i(\mathbf{d},\mathbf{u}) \leq 0 && \forall \mathbf{u} \in U, \forall i \in I_c = [1,...,s]^T \end{aligned} \tag{5.2}$$

The reader finds the generalisation to the CMOP formulated in Eq. (5.1) in Appendix B.

5.2 A Memetic Single Objective Constrained Min-Max Approach

The proposed algorithm is a bi-level optimisation procedure based on the alternation of a minimisation and a restoration step. The minimisation step searches for a global solution to the constrained min-max problem:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}_{af}\in \bar{A}_{uf}} f(\mathbf{d},\mathbf{u}_{af}) \\ &s.t. \\ &\max_{\mathbf{u}_{ac}\in \bar{A}_{uc}} \max_{i\in I_c} c_i(\mathbf{d},\mathbf{u}_{ac}) \leq 0. \end{aligned} \tag{5.3}$$

While the restoration step searches for a solution to the following two global maximisation problems, given the solution $\bar{\mathbf{d}}$ coming from Eq. (5.3):

$$\begin{aligned} &\max_{\mathbf{u}\in U} f(\bar{\mathbf{d}},\mathbf{u}) \\ &s.t. \\ &\max_{i\in I_c} c_i(\bar{\mathbf{d}},\mathbf{u}) \leq 0 \end{aligned} \tag{5.4}$$

$$\max_{\mathbf{u}\in U}\max_{i\in I_c} c_i(\bar{\mathbf{d}},\mathbf{u}) \tag{5.5}$$

The two archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ contain respectively the solutions of Eqs. (5.4) and (5.5). By iteratively alternating the minimisation and restoration steps, one fills the two archives with the maxima found in Eq. (5.4) and Eq. (5.5). Thus we can say that

Eq. (5.3) searches for an optimal $\mathbf{d}$ over a discrete representation of the space of the maxima of objective and constraint functions.

The approach is summarised in Algorithms 1 to 9 and 15 and explained in the following subsections with the help of the flow diagrams in Figs. 5.1 and 5.2 and the examples in Figs. 5.3, B.1 and B.2. In particular, Fig. 5.1 includes both SOP and MOP. The former problem is described in this chapter while the latter is in Appendix B.

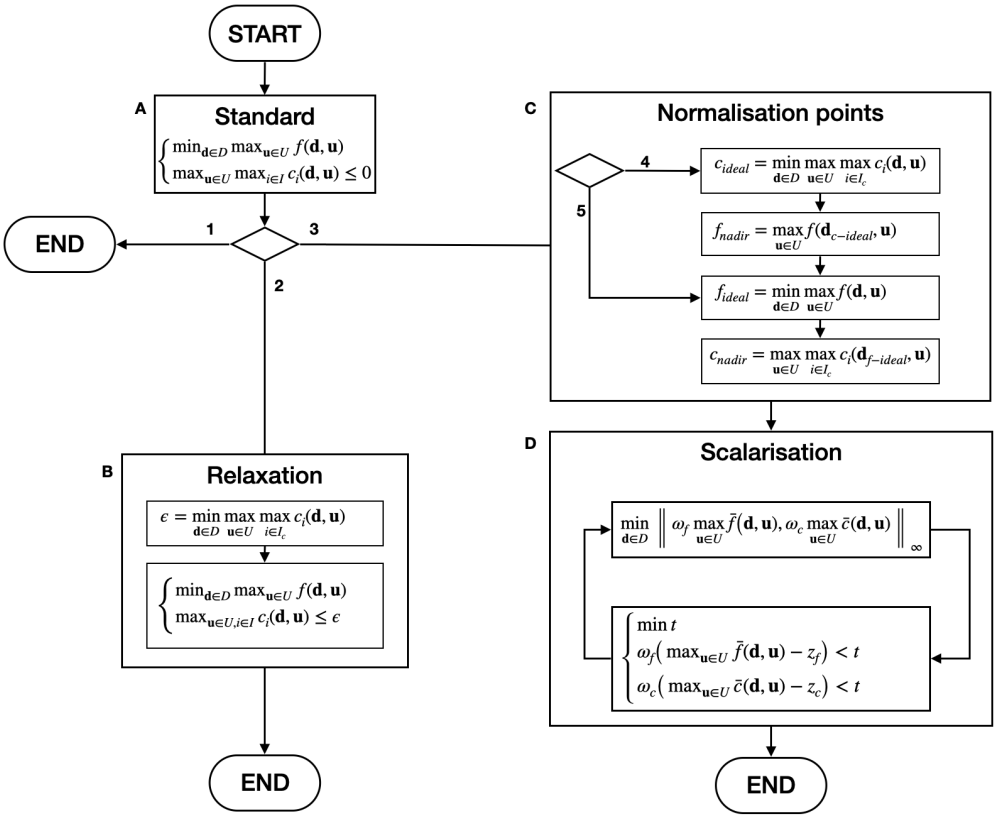


Figure 5.1: Flow diagram of the constrained min-max alternative approaches described in Sections 5.2.1 to 5.2.4 and Appendix B and summarised in Algorithms 1 to 7 and 15. In particular, this diagram describes the sequence of optimisation problems applied for the standard approach and its alternative strategies: the constraint relaxation and the scalarisation. The first optimisation problem is defined in block A and it refers to the general constrained min-max in Eq. (5.1) and Fig. 5.2 . If a solution exists for this problem, no further analysis is required (link 1). If differently, two alternatives are given. The first (link 2) brings to the relaxation strategy (block B). Here an unconstrained min-max problem is solved to find ϵ. The relaxed constrained min-max problem is then solved (Section 5.2.3). Link 3 instead brings to the scalarisation strategy (Appendix B). Block C is first solved to find the reference points in the Pareto front (ideal and nadir). Finally, the scalarisation procedure is activated (block D).

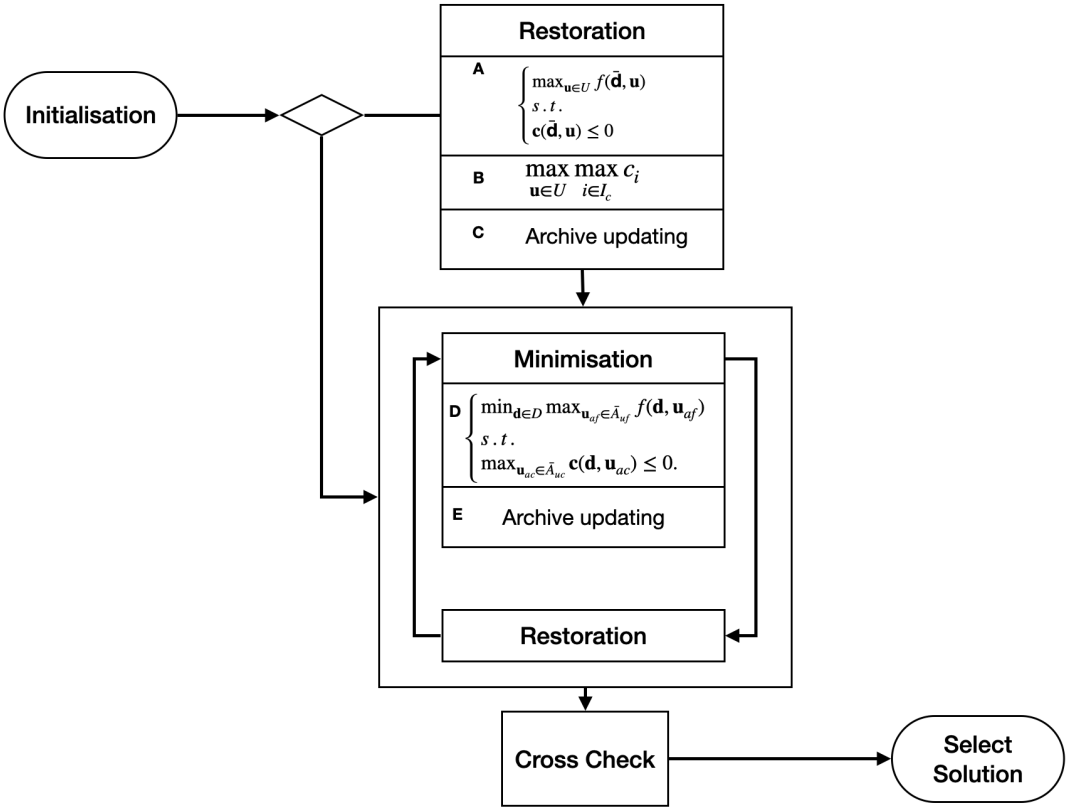


Figure 5.2: Flow diagram of the constrained min-max algorithm for the standard approach in block A in Fig. 5.1. The relaxation and scalarisation alternatives (blocks B,C and D) follow however the same logic. Algorithms 1, 2, 4, 6 and 7 and Sections 5.2.1, 5.2.2 and 5.2.4 explain in detail all the represented blocks. Each of them lists the main operations that are performed: the optimisation problems and the archives updating. The algorithm is initialised. Then there is the iteration between blocks 'Restoration' and 'Minimisation'. Finally the 'Cross Check' is performed and the solution is chosen.

5.2.1 Initialisation

The algorithm is initialised either by selecting a random value $\bar{\mathbf{d}}$ or with a warm start for given first guess and then by searching for a first solution to Eq. (5.4) and Eq. (5.5). These first solutions are then saved into the two archives A_{uf} and A_{uc}. Eq. (5.4) and Eq. (5.5) are global optimisation problems in general. In the following, we will propose the use of a memetic global optimiser which combines DE with a multi-restart mechanism and local search. Given the global nature of the search for a solution, one needs to set the amount of computational resources allocated to the solution of each of the minimisation and maximisation problems. We quantify this resource in terms of objective function evaluations. More specifically, with reference to Eq. (5.4) the maximum number of function evaluations $n_{\text{feval,max}}$ is the number of calls to the objective function $f(\bar{\mathbf{d}},\mathbf{u})$ while for Eq. (5.5) we count the function calls to $\max_i c_i(\bar{\mathbf{d}},\mathbf{u})$.

Once the maximum number of function evaluations per sub-problem is defined one needs to define the maximum number of iterations $n_{\text{loop,max}}$ of minimisation/restoration.

The steps for the initialisation are summarised in Algorithm 1. Line 1 presents how to define the problem in order to be solved by Algorithm 2: the required information on objective function $f(\mathbf{d}, \mathbf{u})$, set of constraint functions $\mathbf{c}(\mathbf{d}, \mathbf{u})$, dimensions n_D and n_U of the design and uncertain vectors $\mathbf{d} \in D \subset \mathcal{R}^{n_D}$ and $\mathbf{u} \in U \subset \mathcal{R}^{n_U}$ and the bounds of each variable in $\mathbf{d}$ and $\mathbf{u}$. Line 2 defines the constraints on the computational resources for the single sub-problems in Eqs. (5.3) to (5.5). They are the maximum number of function evaluations $n^{\text{outer}}_{\text{feval,max}}$ for Eq. (5.3), $n^{\text{inner}}_{\text{feval,max}}$ for Eq. (5.4) and $n^{\text{inner},c}_{\text{feval,max}}$ for Eq. (5.5). Line 3 defines the threshold σ_{stop} below which Algorithm 2 is considered to have converged. Line 4 defines the limit on the computational cost for the whole procedure in Eq. (5.1) with its maximum number of function evaluations $n_{\text{fval,max}}$ and maximum number of iterations $n_{\text{loop,max}}$ between minimisation and restoration. Line 5 defines the optimisation algorithms and its parameter settings that are used to solve the sub-problems in Eqs. (5.3) to (5.5). On line 6 the initial design vector is defined by the user or initialised randomly with a hyper-cube sampling procedure. In the case some important uncertain scenarios in U or design configurations in D are already known, the initial archives can be seeded with these scenarios, otherwise they are initialised as empty sets (line 7). Similarly, it is done for the archives relative to functions f and $\mathbf{c}$ (line 8). Lines 9 and 10 initialise the counters of the number of function evaluation n_{feval} and the iteration n_{loop} between minimisation and restoration processes. The initial accepted constraint violation ϵ (line 11) is initialised to zero.

Algorithm 1 Initialisation

1: Define the problem: objective function f, constraint function c, dimension (n_D and n_U), lower and upper bounds for both design space D and uncertain space U.
2: Define computational limits to the sub-problems: $n^{\text{inner},f}_{\text{fval,max}}$, $n^{\text{inner},c}_{\text{feval,max}}$, $n^{\text{outer}}_{\text{fval,max}}$
3: Define algorithm convergence threshold σ_{stop}.
4: Define computational limits to the whole algorithm: $n_{\text{fval,max}}$ and $n_{\text{loop,max}}$
5: Define optimiser(s) and its (their) parameters setting for all optimisation steps
6: Define/initialise design vector $\bar{\mathbf{d}}$ (latin hypercube sample or input)
7: Define/initialise vectors archives: A_{uf}, A_{uc}, A_d, A_u ($\emptyset$ or input)
8: Define/initialise functions archives: A_f, A_{cf}, A_c ($\emptyset$ or input)
9: Initialise number of function evaluation $n_{\text{feval}} = 0$
10: Initialise number of loops $n_{\text{loop}} = 0$
11: Initialise accepted violation $\epsilon = 0$

5.2.2 Minimisation-Restoration Loop

The main algorithm is summarised in Algorithm 2. Using the first design guess $\bar{\mathbf{d}}$ from the initialisation, Eqs. (5.4) and (5.5) are solved in parallel (lines 4 and 5) within the first restoration, or inner, loop (lines from 2 to 8). With $\bar{\mathbf{d}}$ fixed, the former equation evaluates the feasible worst case condition of f while the latter determines the worst constraint violation. In the following we will adopt a variation of MP-AIDEA [131] to

solve both Eqs. (5.4) and (5.5). The contribution to the algorithm is described in the Section 5.2.5. Since MP-AIDEA is a multi-population procedure, multiple solutions are returned. The solution $\mathbf{u}_{af}$ is selected following Algorithm 3. In particular, if none of the outputs from the optimiser is feasible (no elements in no one of the populations), the uncertain vector with the lowest constraint violation is chosen (lines 2 to 3 of Algorithm 3). Otherwise, within the set of feasible solutions, the uncertain vector with the highest value of f is selected (line 5 of Algorithm 3). The uncertain vectors solutions are then stored in the archives as illustrated in Algorithm 4. Consider first lines 9 to 11 of Algorithm 4: the uncertain vector solutions $\mathbf{u}_{af}$ corresponding to the feasible maximum of f for the fixed $\bar{\mathbf{d}}$ is saved in A_{uf}, the corresponding $f(\bar{\mathbf{d}}, \mathbf{u}_{af})$ and $\max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_{af})$ are saved in A_f and A_{cf} respectively. Consider now lines 12 and 13 of Algorithm 4: the uncertain vector solutions $\mathbf{u}_{ac}$ that maximises the constraint violation for the same fixed design vector is saved in A_{uc}, while the corresponding $\max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_{ac})$ is saved in A_c. Finally, on lines 14 and 15 of Algorithm 4 the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ are created by removing from A_{uf} and A_{uc} all the repeated elements (with a selected tolerance of 1e-8 on the euclidean distance between two elements in the archive).

The main loop is then started (line 9 of Algorithm 2 and Eq. (5.3)) with an alternation of the minimisation, or outer, step (lines 10, 11 and 12) and restoration, or inner, step (lines from 13 to 17). The latter has been already described. A further check is, however, performed in lines 1 to 8 of Algorithm 4 before the updating procedures of the archives. If the condition in line 2 holds, indeed, the inner loop in Eq. (5.4) has failed. Then the solution generated in this loop is discarded and the one of the previous outer loop is maintained. This condition follows the same criterion used in Algorithm 3 by giving priority to the constraint satisfaction. Similarly the condition in line 6 for Eq. (5.5).

The outer step, on the other hand, addresses the solution of the constrained minimisation in Eq. (5.5). For each new design vector generated by the optimiser, the objective and constraint functions are evaluated for all the $\mathbf{u}$ vectors in $\bar{A}_{uf}$ and $\bar{A}_{uc}$ and the worst cost function and constraint violation values are retained (line 11). Note that in some cases it is desirable to run a local search every time a vector in the two archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ is evaluated (see for example [189]). This added local search significantly increases the computational cost of each single evaluation of the outer loop but in some cases improves convergence to the point that the overall cost of the algorithm is reduced. For this reason we inserted this option in the algorithm although it is not tested in this paper. It was, however, tested, for the case of unconstrained min-max problems, in a previous work by the authors [189]. At the end of the minimisation process, the archive A_d of the design configurations is updated with the new design solution $\bar{\mathbf{d}}$.

If the condition in line 18 of Algorithm 2 holds, the constraint relaxation strategy described in Algorithm 5 is activated. An alternative option, the trade-off strategy in Algorithm 15, can instead be activated if condition in line 21 is true. These alternatives can be visualised in Fig. 5.1. If the former condition holds, problem A is stopped, the relaxation step in the first block in B is performed finding ϵ and finally the constrained min-max problem is updated as in the second block in B. If instead the second condition

hold, block A is stopped and then blocks C and D are solved (blocks C and D are related to the MOP and then they are linked to Appendix B).

The minimisation and restoration steps are alternated until condition in line 9 is satisfied. In particular, the iteration holds until the number of calls to the function has not reached the maximum allowed ($n_{\text{feval}} < n_{\text{feval,max}}$), the number of iterations is below the upper bound ($n_{\text{loop}} < n_{\text{loop,max}}$) and the solutions saved in archives have not converged ($\max_A(\sigma) > \sigma_{\text{stop}}$) where $\max_A(\sigma)$ is the maximum standard deviation between all the archives in the last 3 iterations. The violation of at least one of these conditions corresponds to the termination criterion.

Then, the cross-check between all the design vectors archived in A_d is performed (line 26). The cross-check procedure is explained in Section 5.2.4 and summarised in Algorithm 6. Finally, the solution is chosen following Algorithm 7 (line 28).

An example of the application of Algorithm 2 to the two functions MWP10 and GFc1 (see Section 5.4.1), without constraint relaxation and trade-off, can be visualised in Fig. 5.3. In sub-figures (c) and (d) an initial design guess $\hat{\mathbf{d}}$ and each new design $\bar{\mathbf{d}}$ proposed by the minimisation step are represented as vertical lines. The corresponding worst-case scenario for f and $\mathbf{c}$ are plotted, with the same colour, as dots and stars respectively. Sub-figure (e) shows sections of f and $\mathbf{c}$ over U for different design configurations. Sub-figure (f) shows how the space of the maxima of f and $\mathbf{c}$ as it appears in the minimisation process. In particular, sub-figure (c) shows that the algorithm is able to find at the first iteration, for this test case, the subspace $\hat{D} \in D$ in which the design solutions are feasible for any uncertain scenario. The first design guess $\hat{\mathbf{d}}$ (blue line) finds the worst scenario for $\mathbf{u} = 10$. In (b) indeed, the maximum constraint violation is brought to zero at the second iteration and all the other iterations are then used to minimise the worst case of f working within the design domain $\hat{D}$.

5.2.3 Constraint Relaxation Strategy

The *min-max* problem proposed in this paper imposes quite stringent conditions on the satisfaction of the constraints as they need to be satisfied for all possible values of $\mathbf{u} \in U$. It is, therefore, possible that no solution $\mathbf{d}_{\text{opt}}$ is feasible in all U. Since we are interested in the worst case solution for both constraints and objective function, when no feasible $\mathbf{d}$ is possible we introduce an automatic relaxation of the constraints.

The relaxation strategy then translates Eq. (5.2) to a slightly different optimisation problem where the constraint function has to be smaller than ϵ with this value to be determined:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}) \\ &s.t. \\ &c_i(\mathbf{d},\mathbf{u}) \leq \epsilon \qquad \forall \mathbf{u} \in U, \forall i \in I_c = [1,...,s]^T \end{aligned} \tag{5.6}$$

In order to find a set $X \subset D$ that is feasible for all $\mathbf{u} \in U$, we first solve the following minmax problem:

$$\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} \max_{i\in I_c} c_i(\mathbf{d},\mathbf{u}) \tag{5.7}$$

Eq. (5.7) minimises the maximum violation of the constraints and returns a solution

Algorithm 2 Constrained min-max

1: Initialisation: Algorithm 1
2: **if** $\bar{A}_{uf} = \emptyset \wedge \bar{A}_{uc} = \emptyset$ **then**
3: *Initialisation loop*:
4: Run $\mathbf{u}_{af} = \arg\max_{\mathbf{u}\in U} f(\hat{\mathbf{d}}, \mathbf{u})$ s.t. $\max_{i\in I_c} c_i(\hat{\mathbf{d}}, \mathbf{u}) \leq \epsilon$
5: Run $\mathbf{u}_{ac} = \arg\max_{\mathbf{u}\in U} \max_i c_i(\hat{\mathbf{d}}, \mathbf{u})$
6: if multiple outputs, choose best $\mathbf{u}_{af}$ as in Algorithm 3
7: Update archives as in Algorithm 4
8: **end if**
9: **while** $n_{\text{feval}} < n_{\text{feval,max}} \wedge n_{\text{loop}} < n_{\text{loop,max}} \wedge \max_A(\sigma) > \sigma_{\text{stop}}$ **do**
10: *Minimisation loop*:
11: $\bar{\mathbf{d}} = \arg\min_{\mathbf{d}\in D}\{\max_{\mathbf{u}_a\in\bar{A}_{uf}} f(\mathbf{d}, \mathbf{u}_a)\}$ s.t. $\max_{\mathbf{u}_a\in\bar{A}_{uc}} \max_{i\in I_c} c_i(\mathbf{d}, \mathbf{u}_a) \leq \epsilon$
12: Update global archive $A_d = A_d \cup \{\bar{\mathbf{d}}\}$
13: *Restoration loop*:
14: Run $\mathbf{u}_{af} = \arg\max_{\mathbf{u}\in\mathbf{U}} f(\bar{\mathbf{d}}, \mathbf{u})$ s.t. $\max_{i\in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}) \leq \epsilon$
15: Run $\mathbf{u}_{ac} = \arg\max_{\mathbf{u}\in U} \max_i c_i(\bar{\mathbf{d}}, \mathbf{u})$
16: if multiple outputs, choose best $\mathbf{u}_{ac}$: Algorithm 3
17: Update archives: Algorithm 4
18: **if** relaxation flag $\wedge$ not convergence $\wedge$ satisfy limits on n_{feval} and n_{loop} **then**
19: Stop Algorithm 2 and apply the constraint relaxation strategy: Algorithm 5
20: **end if**
21: **if** trade-off flag $\wedge$ convergence on $\mathbf{d}$, $f_{\max}$, $c_{\max}$ $\wedge$ satisfy limits on n_{feval} and n_{loop} **then**
22: Stop Algorithm 2 and apply the trade-off strategy: Algorithm 15
23: **end if**
24: **end while**
25: **for** all $\mathbf{d} \in A_d$ **do**
26: Cross-check: Algorithm 6
27: **end for**
28: Select which solution $[\mathbf{d}_{\text{opt}}, \mathbf{u}_{\text{opt}}]$ to return: Algorithm 7

vectors $\mathbf{d}_{\min,c}$ and $\mathbf{u}_{\min,c}$. Vectors $\mathbf{d}_{\min,c}$ satisfies the constraint:

$$\mathbf{c}_\epsilon \leq 0 \tag{5.8}$$

for all $\mathbf{u} \in U$, where:

$$\mathbf{c}_\epsilon = \mathbf{c} - \epsilon \tag{5.9}$$

and

$$\epsilon = \max_{i\in I_c} c_i(\mathbf{d}_{\min,c}, \mathbf{u}_{\min,c}) \tag{5.10}$$

The relaxation strategy is explained in Algorithm 5. If condition on line 18 of Algorithm 2 is satisfied then Algorithm 5 is triggered. Once in Algorithm 5, until condition in line 2 holds, Eq. (5.7) is solved (see lines 3 to 8 of Algorithm 5), with the iteration

Algorithm 3 Solution Selection - Inner loop

1: **for** all output $\mathbf{u}_{af}$ of the restoration level **do**
2: **if** $\nexists \mathbf{u}_{af} \rightarrow \max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_{af}) \leq \epsilon$ **then**
3: select $\mathbf{u}$ with minimum violation
4: **else**
5: select between the feasible $\mathbf{u}$ the one with the highest objective function *f*.
6: **end if**
7: **end for**

Algorithm 4 Archive Updating

1: **if** in the main loop **then**
2: **if** $(f_{\text{out}} > f_{\text{in}} \wedge c_{\text{out}} < 0) \vee (0 < c_{\text{out}} < c_{\text{in}})$ **then**
3: update $\mathbf{u}_{a,f} = \mathbf{u}$ from outer loop
4: **end if**
5: **if** $c_{\text{out}} > c_{\text{in}}$ **then**
6: $\mathbf{u}_{a,c} = \mathbf{u}$ from outer loop
7: **end if**
8: **end if**
9: $A_{uf} = A_{uf} \cup \{\mathbf{u}_{af}\}$
10: $A_f = A_f \cup \{f(\bar{\mathbf{d}}, \mathbf{u}_{af})\}$
11: $A_{cf} = A_{c,f} \cup \{\max_i c_i(\bar{\mathbf{d}}, \mathbf{u}_{af})\}$
12: $A_{uc} = A_{uc} \cup \{\mathbf{u}_{ac}\}$
13: $A_c = A_c \cup \{\max_i c_i(\bar{\mathbf{d}}, \mathbf{u}_{ac})\}$
14: $\bar{A}_{uf} = A_{uf} \setminus$ repeated solutions
15: $\bar{A}_{uc} = A_{uc} \setminus$ repeated solutions

between the following minimisation

$$\min_{\mathbf{d} \in D} \max_{\mathbf{u}_{ac} \in \bar{A}_{uc}} \max_{i \in I_c} c_i(\mathbf{d}, \mathbf{u}_{ac}), \tag{5.11}$$

and restoration step

$$\max_{\mathbf{u} \in U} \max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}) \tag{5.12}$$

This is an unconstrained min-max formulation where the optimised function is the vector of constraints $\mathbf{c}$. The solution at convergence is the minimum over D of the worst constraint violations in U. In line 10 that value is associated to the relaxation parameter ϵ.

5.2.4 Cross-Check

All the optimisation problems in the minmax algorithm require the identification of a global maximum or a global minimum. Since it is proposed to use a memetic algorithm it is possible that some of the maxima or minima in the archive are only locally optimal. Note that the use of a deterministic global optimiser would remove this problem but

Algorithm 5 Constraint Relaxation

1: Inherit vectors from Algorithm 2
2: **while** not convergence of relaxation **do**
3: *Minimisation loop*:
4: $\bar{\mathbf{d}} = \arg\min_{d\in D}\{\max_{\mathbf{u}_{ac}\in \bar{A}_{uc}} \mathbf{c}(\mathbf{d}, \mathbf{u}_{ac})\}$
with the cross-check as in Algorithm 6.
5: *Restoration loop*:
6: Run $\mathbf{u}_{ac} = \arg\max_{\mathbf{u}\in U} \max_i c_i(\bar{\mathbf{d}}, \mathbf{u})$
7: if multiple outputs, choose best $\mathbf{u}_{ac}$: Algorithm 3
8: Update archives A_{uc} and A_c: Algorithm 4
9: **end while**
10: set $\epsilon = \max_{\mathbf{u}\in U} \max_i c_i(\bar{\mathbf{d}}, \mathbf{u})$
11: Restart Algorithm 2

would introduce a tractability problem due to the potential NP-hard nature of some optimisation problems.

In order to mitigate the occurrence of local minima/maxima in the archives we introduce a cross-check of the solutions following the procedure explained in Algorithm 6. It is performed for each design vector $\bar{\mathbf{d}}$ that can be proposed by the optimiser during the minimisation step and at the end of the whole algorithm (respectively in line 9 and 19 of Algorithm 2). Referring to Algorithm 6, lines 1 to 7 regard the objective function f while lines from 8 to 14 regard the constraint function $\mathbf{c}$. In both cases, for a given $\bar{\mathbf{d}}$ objectives and constraints are evaluated for all the $\mathbf{u}$ vectors in the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$. We also introduced an option (through *local flag*) to run a local search from each new pair [$\bar{\mathbf{d}}$, $\mathbf{u}$]. This option slows down Algorithm 2 but improves the quality of the solution if the functions present nested minima/maxima. Finally, line 15, retains the worst values of f and $\mathbf{c}$ for the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ for each $\bar{\mathbf{d}}$.

After the termination criterion in Algorithm 2 is applied and the cross-check over the archives is performed (line 21), the solution for the min-max problem is selected following Algorithm 7. In particular, if a feasible subset $\hat{A}_d$ of the archive A_d of the design vectors exists (line 1) the selected solution vector is the one, within $\hat{A}_d$, minimising the worst value of f (line 2). If, on the other hand, $\hat{A}_d$ is an empty set, the design vector that minimises the constraint violation is selected (line 4).

5.2.5 A Memetic Strategy for Constrained Global Optimisation Problems

The whole procedure for the solution of the global min-max problem is agnostic to the optimisation solver used for the single problems in Eqs. (5.3) to (5.5) within Algorithm 2. However, when problems in Eqs. (5.3) to (5.5) are global optimisation problems we propose the use of a generalisation of the memetic optimisation algorithm MP-AIDEA [131]. This section briefly explains the contribution done in the generalisation to the handling of constraints while the generalisation to MOP is explained instead in Appendix B. The optimisation problem we are interested to solve is the

Algorithm 6 Cross-Check

for all elements $\mathbf{u}_{af} \in \bar{A}_{uf}$ **do**
  if local flag **then**
    Compute local maximum $f(\bar{\mathbf{d}}, \mathbf{u}_a^*)$ s.t. $\max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_a^*) \leq \epsilon$ from $\mathbf{u}_{af}$
  else
    Compute $f(\bar{\mathbf{d}}, \mathbf{u}_a)$ s.t. $\max_{i \in I_c} c_i(\mathbf{d}, \mathbf{u}_{af}) \leq \epsilon$
  end if
end for
for all elements $\mathbf{u}_{ac} \in \bar{A}_{uc}$ **do**
  if local flag **then**
    Compute local maximum $\max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_a^*)$ from $\mathbf{u}_{ac}$
  else
    Compute $\max_{i \in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}_{ac})$
  end if
end for
For each $\bar{\mathbf{d}}$ save worst vectors $\mathbf{u}_{af}$ and $\mathbf{u}_{ac}$ in the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$.

Algorithm 7 Select Solution - Output

if $\hat{A}_d = \{\mathbf{d} | \max_{\mathbf{u}_{ac} \in A_{uc}} \max_{i \in I_c} c_i(\mathbf{d}, \mathbf{u}_{ac}) \leq \epsilon\} \neq \emptyset$ **then**
  take $\mathbf{d} \in \hat{A}_d$ that minimise $\max_{\mathbf{u}_{af} \in A_{uf}} f(\mathbf{d}, \mathbf{u}_{af})$
else
  take $\mathbf{d} \in A_d$ that minimise $\max_{\mathbf{u}_{ac} \in A_{uc}} \max_{i \in I_c} c_i(\mathbf{d}, \mathbf{u}_{ac})$
end if

general CSOP minimisation problem:

$$\begin{aligned} &\min_{\mathbf{x} \in X} \phi(\mathbf{x}) \\ &s.t. \\ &\gamma_i(\mathbf{x}) \leq \epsilon \quad \forall i \in I_c = [1, ..., s]^T \end{aligned} \tag{5.13}$$

where Eq. (5.13) incorporates the three distinct Eqs. (5.3) to (5.5) and the mapping is explained in the following of the section.

The memetic solver combines the Darwinian evolution of a number of populations of candidates through a DE strategy with the Lamarckian evolution of the best agent in each converged population through a local search algorithm. In particular, the local refinement is performed if the converged solution is not in the basin of attraction of previous local minima. A number of local restarts are then performed before globally restarting. The general structure of the optimisation solver for CSOP is summarised in Algorithm 8.

First there is the initialisation of algorithm's parameters: the maximum number of function evaluation $n_{\text{feval,max}}$, the number of populations n_{pop}, the number of agents in each population N_{pop}, the convergence threshold ρ of DE and the radius of the global restart bubble δ_{global}. n_{LR} and δ_{local} can also be defined or, alternatively, auto-adapted

Algorithm 8 MP-AIDEA (Constraint handling)

1: Initialisation: $n_{\text{feval,max}}$, n_{pop}, N_{pop}, ρ, δ_{global} (optional: n_{LR} δ_{local}).
2: **while** $n_{\text{feval}} < n_{\text{feval,max}}$ **do**
3: Run the DE step (Algorithm 9)
4: **for** $p \in [1, 2, ..., N_{\text{pop}}]$ **do**
5: **if** $\mathbf{x}_{p,\text{best}}$ not in the basin of attraction of previous solutions **then**
6: Run local search
7: update $\mathbf{x}_{p,\text{best}}$ from the local search.
8: **end if**
9: **end for**
10: Initialise populations for local or global restart in the next DE step.
11: **end while**

Algorithm 9 DE step (Constraint handling)

1: **for** $p \in [1, 2, ..., N_{\text{pop}}]$ **do**
2: *Initialise* at generation $G = 1$ the genotype $\mathbf{x}_{p,q}^{(G)}$ $\forall$ p-agents and q-populations
3: *Evaluate* the phenotype of each candidate solution: $f_{p,q}^{(G)}$ Eq. (5.14)
4: **while** the population is not contracted **do**
5: *Select* parents: all generation G;
6: *Variate* the parent's genotype: two strategies randomly alternated (DE/Rand/1/bin, DE/CurrentToBest/2/bin)
7: *Evaluate* new candidates $f_{p,q}^{(G+1)}$ Eq. (5.14).
8: *Select* between parents and children with a greedy criterion
9: update generation: G = G+1.
10: **end while**
11: return $\mathbf{x}_{p,\text{best}}$;
12: return $f(\mathbf{x}_{p,\text{best}})$;
13: **end for**

by the solver.

The optimisation process then hybridises the DE step (line 3) where the N_{pop} populations are evolved and the local search (line 6) where their best candidate solutions are refined. The number of local refinements is adapted within MP-AIDEA allowing them to be run only if the converged solution in the DE is outside the basins of attraction of the previous recorded local minima which depend on the distances between previous best solutions of the DE and best solutions of the local search. The DE is then locally and globally restarted (line 10) until the maximum number of evaluations $n_{\text{feval,max}}$ (considering both DE and local search) of the objective function is achieved (termination condition in line 2).

More details about the DE step for the CSOP are in Algorithm 9. Following the building blocks that make any EA, the DE step can be divided in: initialisation (line 2), variation (line 6), evaluation (lines 3 and 7), selection (line 8) and termination (line 4). Within the main loop (lines 4-10) all the agents at the current generation

G are selected as parents and are subjected to the variation step for the definition of generation G+1. The two schemes DE/Rand/1/bin and DE/CurrentToBest/2/bin [190] have been implemented for the parent's variation. The best for each agent between the corresponding parent at generation G and offspring at generation G+1 is finally selected.

An important aspect of the optimisation solver is the handling of constraints. This is done at the level of function evaluation and explained in Section 5.2.5.

An example of a run of MP-AIDEA with $n_{\text{pop}} = 2$ for problem *MWP-1&GFc-1* is in Fig. 5.4 which shows, for the evolving populations, the alternations of DE steps together with the local refinements. For this test case the MATLAB function solver *fmincon* has been used for the local search. Given the constraint on the maximum number of function evaluations, the algorithm is able to perform only one local restart in each population. Colours red, blue and green refer to the first population while colours orange, yellow and brown refer to the second population. As shown in the figure, first the two populations are initialised randomly and evolved independently until convergence is achieved (being the parameter ρ one of the termination criteria). In particular the red (orange) area represent the evolution of the best agent in the population while the blue (yellow) represent the mean value and the green (brown) the worst agent. From the two converged solutions a local search is performed with *fmincon* until the green and brown points are obtained. From these two local minima then the two populations are locally reinitialised. The overall process is then repeated till convergence.

Constraint Handling

It is here described the approach to handle the constraints as in line 6 of Algorithm 8. Since the optimisation solver is a memetic approach, the constraint handling procedure needs to be included in both global evolutionary exploration and local exploitation.

Regarding the former, the DE step, and with reference to [36, 39] we propose the following indirect approach with an adaptive exterior penalty function for hard constraint handling where hardness refers to the absolute satisfaction of the constraint. By 'indirect approach' we mean that the COP is translated to a FOP: this type of constraint handling is applied within the evaluation step in DE: line 7 of Algorithm 9. The following mapping is used:

$$f(\mathbf{x}_{p,q}) = \begin{cases} f(\mathbf{x}_{p,q}) & \text{if } \max_i c_i(\mathbf{x}_p) \leq 0 \\ \max_q\{f_s(\mathbf{x}_{p,q})\} + \max_j\{c_j(\mathbf{x}_{p,q})\} & \text{else} \end{cases} \tag{5.14}$$

where, for a generic population f is the value of the objective function for the given agent q in the population p, $\max_q\{f(\mathbf{x}_{p,q})\}$ is the maximum of f over the current population and $\max_j\{c_j(\mathbf{x}_{p,q})\}$ is the maximum constraint violation for the considered agent q.

For the local search (line 6 of Algorithm 8) instead the constrained formulation in Eq. (5.13) is directly handled within the nonlinear programming solver `fmincon`.

5.3 Computational Complexity

The computational cost of Algorithm 2 is measured in terms of number of function calls. With reference to the minimisation step, the counter $n_{\text{feval}}^{\text{outer}}$ takes into account, for both the constrained and the unconstrained min-max problems, the calls to $\max_{\mathbf{u}\in\bar{A}_{uf}} f(\mathbf{d},\mathbf{u}_{af})$ in Eq. (5.3). The same criterion, then, holds for the constraint relaxation step in Eq. (5.11) and for the two trade-off steps in Eqs. (B.6) and (B.17). It has to be noted that, as the algorithm proceeds in the search of the global optimum solution, the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ of the uncertainty vectors increase progressively in dimension. Each minimisation step explores a maximum number of possible design configurations which is limited by the input parameter $n_{\text{feval,max}}^{\text{outer}}$. However, due to the growth of the archives of the solutions coming from the restoration step, each evaluation of the minimisation loop becomes increasingly more expensive.

With reference to the maximisation step, instead, the cost of the two separate problems in Eq. (5.4) and Eq. (5.5) has to be considered. For the former $n_{\text{feval}}^{\text{inner},f}$ counts the number of calls to the objective function $f(\bar{\mathbf{d}},\mathbf{u})$ and it is limited by the input $n_{\text{feval,max}}^{\text{inner},f}$. This holds true also for the two steps in the trade-off strategy in Eqs. (B.11) and (B.12) and for the relaxation step in Eq. (5.12) where the function c is considered instead of f. For the latter, $n_{\text{feval}}^{\text{inner},c}$ counts the number of calls to $\max_i c_i(\hat{\mathbf{d}},\mathbf{u})$ in Eq. (5.5) where the input $n_{\text{feval,max}}^{\text{inner},c}$ is the upper limit.

Finally, the parameters $n_{\text{feval,max}}$ and $n_{\text{loop,max}}$ give an upper limit on the whole cost of Algorithm 2. Note that $n_{\text{feval,max}}$ and $n_{\text{loop,max}}$ represent an upper limit because, as it is shown in line 9 of Algorithm 2, we use an additional termination criterion that looks at the convergence of the solutions in the archives.

The computational complexity of the different parts of the overall algorithm is as follows:

1. **Local Search**: the local search uses the Matlab *fmincon* function. All the alternative algorithms (*interior-point*, *trust-region-reflective*, *sqp*, *sqp-legacy*, *active-set*) can be selected. We use here *interior-point* that works well with both large sparse and small dense problems. The complexity is $\mathcal{O}(n_D^3)$ or $\mathcal{O}(n_U^3)$ depending on which step between minimisation and restoration is considered, where n_D is the design and n_U the uncertain vector's dimension.

2. **Adaptation of CR and F**: The DE parameters CR and F in MP-AIDEA are auto-adapted for each element of each population. For n_{pop} evolving populations with N_{pop} agents, the complexity is $\mathcal{O}(n_{\text{pop}}N_{\text{pop}}n_D^2)$ and $\mathcal{O}(n_{\text{pop}}N_{\text{pop}}n_U^2)$ for inner and outer problem respectively [131].

3. **Restart mechanisms**. The populations evolve with a DE approach which is restarted, locally and globally, a number of times. The local restart has a cost proportional to $n_{\text{pop}}N_{\text{pop}}$. The cost for the global restart, instead, has a component related to the clustering procedure $\mathcal{O}(N_{\text{pop}}n_D n_{\text{iter}})$ or $\mathcal{O}(n_{LM}n_U^2 n_{\text{iter}})$ with n_{iter} the required number of iteration for the clustering and n_{LM} the number of local minima, and a component related to the verification that the new population is far from the clusters $\mathcal{O}(N_{\text{pop}}n_{LM})$ [131].

4. **Outer-Loop**. During the minimisation step there is a cost related to the testing of each design vector suggested by the optimiser in combination with all the uncertain vectors saved in the archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ and there is a cost due to the selection of their maximum. In both cases the complexity is $\mathcal{O}(||\bar{A}_{uf}||n^{\text{outer}}_{\text{fval,max}}) \leq \mathcal{O}(n_{\text{loop}}n^{\text{outer}}_{\text{fval,max}})$ for the objective function f and $\mathcal{O}(||\bar{A}_{uc}||n^{\text{outer}}_{\text{fval,max}}) \leq \mathcal{O}(n_{\text{loop}}n^{\text{outer}}_{\text{fval,max}})$ for the constraint function c.

5. **Cross-Check**. As in the outer loop there is here a cost for the cross-check and a cost for the selection of the maxima. In both cases it is: $\mathcal{O}(||A_d||||\bar{A}_{uf}||) \leq \mathcal{O}(n^2_{\text{loop}})$ and $\mathcal{O}(||A_d||||\bar{A}_{uc}||) \leq \mathcal{O}(n^2_{\text{loop}})$ for f and c respectively because each design vector $\mathbf{d} \in A_d$ is considered.

6. **Select Solution**. After the final cross-check, the archives are updated and the set of design vectors $\hat{A}_d$ feasible in all the uncertain domain can be defined. The min-max solution is selected following Algorithm 7 sorting the feasible solutions f ($\mathcal{O}(||\hat{A}_d||)$) or minimising the constraint violation c ($\mathcal{O}(||A_d||)$).

5.4 Testing Procedure

Algorithm 2 has been tested on the benchmark described and explained in this section. Each test case is a combination of an objective function f and a constraint function c. Depending on the mathematical features of each problem, a local optimiser or a global optimiser have been used for the three problems in Eqs. (5.3) to (5.5). The criteria used to choose the right optimiser is explained in Section 5.4.3.

Given the stochastic nature of MP-AIDEA, each optimisation for each problem has been repeated 100 times. Results are then reported in Section 5.5. For the evaluation of the algorithm's performance, the Success Rate (SR) is used instead of the best value, mean, and variance. The SR was suggested in [129] for a generic problem $\min_f$ and a generic algorithm. It is here generalised to consider also the handling of constraints. The definition of SR is in Section 5.4.2.

5.4.1 Benchmark

The equations of f and c are listed in Tables 5.1 and 5.2 respectively. The constraint functions c are more extensively presented in the following Section 5.4.1 and visualised in Figs. 5.5 to 5.11 for the case $n_D = n_U = 1$. Table 5.3 lists lower and upper bounds, dimensions and optimal solutions for the unconstrained problems in Table 5.1. The same solutions holds also for the constraint min-max problems for which the constraint does not change the global optimum. Table 5.4 presents instead the reference solutions for the constraint min-max problems for which the constraint function changes the position of the global optimum.

MWP-1,2,...,7 are convex-concave test functions taken from chapter 5 of [191]. Objective functions *MWP-8,...,11* are first introduced in [166] and then used also in [42, 157, 158] while *MWP-12,13* are instead selected from [42, 157, 158]. They have been used all together in [43] as benchmark for the unconstrained min-max problem. Functions *GFf-1,2* and *GFc-1,2,...,8* have been specifically designed for the testing of

Algorithm 2, given the lack of a benchmark in the literature for the constrained version of the min-max problem.

Both f and c are designed to include different structures that can be encounter in practice [192]. In particular, they exhibits the following features:

1. **Modality**: number of local optima that try to trap the algorithm in the wrong peak.
2. **Basin** or plateau: a relatively steep decline that surrounds a large area. There is no information to drive the algorithm.
3. **Valley**: similar to the basin but it is narrow area.
4. **Non separability**: property related to the coupling between parameters. Non-separable functions are in general more difficult to optimise.
5. **Dimensionality**: property related to the number of parameters or dimension of the problem. The search space increases with the dimension, increasing then also its difficulty.
6. **Non differentiability**: cuspids, corners, tangents and discontinuities are features that make functions non differentiables in some points. Some of the constraint functions present cuspids, corners and discontinuity. In particular discontinuity is an abrupt change in the function values. Discontinuities are classified in jump, infinite, removable, endpoint, or mixed. Some of the constraint functions c present jump discontinuities.

MWP-9 is not differentiable. *MWP-1,8,9,12,13* are uni-modal in both D and U. On the other hand *MWP-4,5,6,7,10,11* are multi-modal in both D and U. *MWP-2,3* are multi-modal in D only. The new test cases $GFf\text{-}1, 2$ and $GFc\text{-}1, ..., 8$ are explained in the following. They depend on the components d_i of the design vector $\mathbf{d}$, the components u_i of the uncertainty vector $\mathbf{u}$ and the combined vector $\mathbf{x} = [\mathbf{d}, \mathbf{u}]^T$.

GFf-1

GFf-1 is a modifications of the Rastrigin function where half of the variables are design parameters and the others are uncertain variables.

$$c(\mathbf{d}, \mathbf{u}) = 10(n_D + n_U) + \sum_{i=1}^{n_D} (d_i^2 + u_i^2 - 10\big[\cos(2\pi d_i) + \cos(2\pi u_i)\big] - 5 \tag{5.15}$$

It is continuous, differentiable, scalable, without valleys and basins, and highly multi-modal with hundreds of local peaks.

GFf-2

GFf-2 is a variation of the saddle-point function *MWP-8*:

$$c(\mathbf{d}, \mathbf{u}) = \sum_{k=1}^{n} [(d_{k,\mathcal{R}} - 5)^2 - ((u_{k,\mathcal{R}} - 5)^2] \tag{5.16}$$

where both components $d_{k,\mathcal{R}}$ and $u_{k,\mathcal{R}}$ are obtained rotating d_k and u_k respectively by the angle

$$\theta_k = \begin{cases} \pi/8 + 1^k d_{k+1}/20 + u_{k+1}/20 & \text{if } k < n \\ \pi/8 + 1^k d_k/20 + u_k/20 & \text{else} \end{cases} \tag{5.17}$$

GFf-2 is continuous, differentiable, non-separable, scalable, without valleys and basins, and uni-modal.

GFc-1

GFc-1 is a hyper-plane and it is a linear function in both $\mathbf{d}$ and $\mathbf{u}$:

$$c(\mathbf{d}, \mathbf{u}) = \sum_{i=1}^{n} d_i + u_i + K \tag{5.18}$$

where $K = -\sum_i d_i - \sum_i u_{u,i} - 0.05$ with $u_{u,i}$ the upper bound for the i-th uncertain variable. *GFc-1* is continuous, differentiable, separable, scalable, without valleys and basins, and uni-modal.

GFc-2

GFc-2 is a modification of *GFc-1*. It is a continuous piece-wise linear function where the feasible region is a plateau. It is the intersection of two hyper-planes, the second being at the border between of feasible and infeasible regions.

$$\max \left[0, \sum_{i=1}^{n} d_i + u_i + K\right] \tag{5.19}$$

with K as in Eq. (5.18). *GFc-2* is continuous, non differentiable, separable, scalable, with a plateau, without valleys and uni-modal.

GFc-3

In *GFc-3* there are a jump discontinuities, valleys and plateaus. The feasible area is a narrow multidimensional circle. The function is not differentiable, scalable and uni-modal:

$$c(\mathbf{d}, \mathbf{u}) = \begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 1 & \text{else} \end{cases} \tag{5.20}$$

GFc-4

GFc-4 is a modification of the Rastrigin function where a jump discontinuity is introduced:

$$c(\mathbf{d},\mathbf{u}) = \begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 30n + \sum_{i=1}^{n}[x_i^2 - 30\cos(2\pi x_i)] - 30 & \text{else} \end{cases} \tag{5.21}$$

It is highly multi-modal, discontinuous, not differentiable with valleys and plateaus, separable and scalable.

GFc-5

GFc-5 is a modifications of Eq. (5.21). Here a rotation of the vectors $\mathbf{d}$ and $\mathbf{u}$ is also introduced.

$$c(\mathbf{d},\mathbf{u}) = \begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 30n + \sum_{k=1}^{n}[x_{k,\mathcal{R}}^2 - 30\cos(2\pi x_{k,\mathcal{R}})] - 30 & \text{else} \end{cases} \tag{5.22}$$

The rotated components $d_{k,\mathcal{R}}$ and $u_{k,\mathcal{R}}$ are given by the angle $\theta_k = d_i + 2u_i$.

GFc-5 is discontinuous, not differentiable, with valleys and plateaus, scalable, separable and multi-modal.

GFc-6

GFc-6 is a multi-dimensional peak function with high coupling between D and U. It is unfeasible in most of the domain while it is satisfied only in few narrow non linear valleys varying with $\mathbf{d}$.

$$c(\mathbf{d},\mathbf{u}) = \big[\sum_{i=1}^{2n}(x_i - x_i^{opt})\sum_i(d_i^2 - 2u_i)\sum_i(d_i - d_i^{opt})\big]^{(2/7)} \tag{5.23}$$

It presents very narrow non-linear valleys. It is continuous, locally non differentiable, without plateaus, scalable, non separable and multi-modal.

GFc-7

GFc-7 is a multi-dimensional peak functions with high coupling between D and U and narrow unfeasible areas varying with $\mathbf{d}$.

$$\begin{aligned} A &= \textstyle\sum_{i=1}^{n_D}((3/2d_i - (d_{u,i} - d_{opt,i})/2)(\sin(d_i)/d_i) - d_{opt,i}/2)^{1/5} + \\ &\quad \textstyle\sum_{i=1}^{n_U}((3/2u_i - (u_{u,i} - u_{opt,i})/2)(1 - u_{opt,i}/2)^{1/5} \\ B &= \textstyle\sum_{i=1}^{n_D}(d_u(\sin(d_{u,i})/d_{u,i}) - d_{opt,i}/2)^{1/5} + \\ &\quad \textstyle\sum_{i=1}^{n_U}(u_u(1 - u_{opt,i}/2)^{1/5} \\ C &= (d_1 - d_{opt,1})^{2/5} \\ D &= (d_{u,1} - d_{opt,1})^{2/5} \\ c(\mathbf{d},\mathbf{u}) &= 1 - AC/BD \end{aligned} \tag{5.24}$$

where $d_{u,i}$ and $u_{u,i}$ are the upper bounds for d_i and u_i respectively. *GFc-7* has very narrow non-linear peaks with large plateaus. It is continuous, locally non differentiable (for the cuspids), without valleys, scalable, separable and multi-modal in D.

GFc-8

GFc-8 is a rotated versions of *MWP-8*:

$$c(\mathbf{d},\mathbf{u}) = \sum_{i=1}^{n}[(d_{i,\mathcal{R}}-5)^2 - (u_{i,\mathcal{R}}-5)^2] \tag{5.25}$$

where $d_{i,\mathcal{R}}$ and $u_{i,\mathcal{R}}$ are obtained from d_i and u_i respectively with the rotation angle in Eq. (5.17). It is non-separable, scalable, continuous, differentiable and uni-modal and without valleys and plateaus.

Table 5.1: test cases for the objective function f

ID	objective functions
MWP-1	$5(d_1^2+d_2^2)-(u_1^2+u_2^2)+d_1(-u_1+u_2+5)+d_2(u_1-u_2+3)$
MWP-2	$4(d_1-2)^2-2u_1^2+d_1^2u_1-u_2^2+2d_2^2u_2$
MWP-3	$d_1^4u_2+2d_1^3u_1-d_2^2u_2(u_2-3)-2d_2(d_1-3)^2$
MWP-4	$-\sum_{i=1}^{3}(u_1-1)^2+\sum_{i=1}^{2}(d_1-1)^2+u_3(d_2-1)+u_1(d_1-1)+u_2d_1d_2$
MWP-5	$-(d_1-1)u_1-(d_2-2)u_2(d_3-1)u_3+2d_1^2+3d_2^2d_3^2$
MWP-6	$u_1(d_1^2-d_2+d_3-d_4+2)+u_2(-d_1+2d_2^2-d_3^2+2d_4+1)+$ $d_3(2d_1-d_2+2d_3-d_4^2+5)+5d_1^2+4d_2^2+3d_3^2+2d_4^2-\sum_{i=1}^{3}u_i^2$
MWP-7	$2d_1d_5+3d_4d_2+d_5d_3+5d_4^2+5d_5^2-d_4(u_4-u_5-5)+$ $d_5(u_4-u_5+3)+\sum_{i=1}^{3}(u_i(d_i^2-1))-\sum_{i=1}^{5}u_i^2$
MWP-8	$(d_1-5)^2-(u_1-5)^2$
MWP-9	$\min(3-0.2d_1+0.3u_1, 3+0.2d_1-0.1u_1)$
MWP-10	$\frac{\sin(d_1-u_1)}{\sqrt{d_1^2+u_1^2}}$
MWP-11	$\frac{\cos(\sqrt{d_1^2+u_1^2})}{\sqrt{d_1^2+u_1^2}+10}$
MWP-12	$100(d_2-d_1^2)^2+(1-d_1)^2-u_1(d_1+d_2^2)-u_2(d_1^2+d_2^2)$
MWP-13	$(d_1-2)^2+(d_2-1)^2+u_1(d_1^2-d_2)+u_2(d_1+d_2-2)$
GFf-1	$10(n_D+n_U)+\sum_{i=1}^{n_D}(d_i^2+u_i^2-10\big[\cos(2\pi d_i)+\cos(2\pi u_i)\big]-5$
GFf-2	$\sum[(\mathcal{R}_{d_{i+1},u_{i+1}}(d_i)-5)^2-((\mathcal{R}_{d_{i+1},u_{i+1}}(u_i)-5)^2]$

5.4.2 Success Rate

The SR is adopted here for the performance assessment of Algorithm 2. Its definition is given in Algorithm 10 for a generic algorithm $\mathcal{A}$ applied to a generic constrained problem CP on the $D \times U$ space. It is defined as the ratio $\frac{j_s}{n}$ between the index of performance j_s and the number of independent experiments n.

First, all the parameters required by Algorithm 2 are fixed (refers to the initialisation in Algorithm 1). The following parameters are then defined: the number of repetition

Table 5.2: test cases for the constraint functions c

name	constraint functions
GFc-1	$\sum d_i + u_i + K$
GFc-2	$\max\left[0, \sum d_i + u_i + K\right]$
GFc-3	$\begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 1 & \text{else} \end{cases}$
GFc-4	$\begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 30n + \sum[x_i^2 - 30\cos(2\pi x_i)] - 30 & \text{else} \end{cases}$
GFc-5	$\begin{cases} 0 & \text{if } \max_i(d_i - d_i^{opt}) \leq 0.1 \\ 30n + \sum[x_{i,\mathcal{R}}^2 - 30\cos(2\pi x_{i,\mathcal{R}})] - 30 & \text{else} \end{cases}$
GFc-6	$\left[\sum(x_i - x_i^{opt}) \sum(d_i^2 - 2u_i) \sum(d_i - d_i^{opt})\right]^{(2/7)}$
GFc-7	$1 - AC/BD$
GFc-8	$\sum[(d_{i,\mathcal{R}} - 5)^2 - (u_{i,\mathcal{R}} - 5)^2]$

of the experiments n, the tolerances tol_f, tol_d and tol_u on the solution error for the objective function f, the design vector $\mathbf{d}$ and the uncertain vector $\mathbf{u}$ respectively. The formula for SR is in line 15. It depends on the tolerances and on the errors δ_c^k, δ_f^k, δ_d^k and δ_u^k with respect to the reference solutions f_{ref}, $\mathbf{d}_{\text{ref}}$ and $\mathbf{u}_{\text{ref}}$ in Tables 5.3 and 5.4. In particular, δ_c^k depends on the uncertain vector $\mathbf{u}_{\text{opt},c}$ that is the worst for the constraint function c while δ_f^k depends on the vector $\mathbf{u}_{\text{opt},f}$ that makes worst the objective functions f. δ_u is necessary to verify the convergence on the maximisation in the inner loop (restoration in Section 5.2) and then to avoid counting as success solution an f_{opt}^k close to f_{ref} that is coming from a lucky combination of a wrong maximisation and a wrong minimisation in the outer loop (optimisation in Section 5.2).

5.4.3 Algorithm Settings

An important feature of the proposed approach is its modularity in the sense that any optimiser can be plugged in and used for the single optimisation problem in Eqs. (5.3) to (5.5). To enhance efficiencies of Algorithm 2, then, the right combination of optimisation solvers should be selected. An optimal choice would require a prior knowledge of the main features of a given problem. For complex multi-modal functions, we suggest the use of the modified memetic optimisation solver MP-AIDEA because it has shown to be efficient and effective, on average, on a wide range of problems mixing different characteristics. For continuous uni-modal functions we use instead the Matlab `fmincon` solver with an interior-point scheme. We give here the parameter settings of MP-AIDEA that have been used for all tests. The number of agents for each population N_{pop} and the maximum number of function evaluations were set to be respectively $N_{\text{pop}} = \max[5, n_D]$, $n_{\text{feval,max}}^{\text{outer}} = 500 n_D$, $n_{\text{feval,max}}^{\text{inner,f}} = 500 n_U$ and $n_{\text{feval,max}}^{\text{inner,c}} = 500 n_U$. The dimension of the bubble for the global restart is $\delta_{\text{global}} = 0.1$, the number of populations

Table 5.3: Reference solutions for the test cases in table Table 5.1

ID	D	U	d min-max	u min-max	f min-max
MWP-1	$[-5; 5]^2$	$[-5; 5]^2$	-0.4833	0.0833	-1.6833
			-0.3167	-0.0833	
MWP-2	$[-5; 5]^2$	$[-5; 5]^2$	1.6954	0.7186	1.4039
			-0.0032	-0.0001	
MWP-3	$[-5; 5]^2$	$[-3; 3]^2$	-1.1807	2.0985	-2.4688
			0.9128	2.666	
MWP-4	$[-5; 5]^2$	$[-3; 3]^3$	0.4181	0.709	-0.1348
			0.4181	1.0874	
				0.709	
MWP-5	$[-5; 5]^3$	$[-1; 1]^3$	0.1111	0.4444	1.345
			0.1538	0.9231	
			0.2	0.4	
MWP-6	$[-5; 5]^4$	$[-2; 2]^3$	-0.2316	0.6195	4.543
			0.2228	0.3535	
			-0.6755	1.478	
			-0.0838		
MWP-7	$[-5; 5]^5$	$[-3; 3]^5$	1.4252	0.5156	-6.3509
			1.6612	0.8798	
			1.2585	0.2919	
			-0.9744	0.1198	
			-0.7348	-0.1198	
MWP-8	[0; 10]	[0;10]	5	5	0
MWP-9	[0; 10]	[0;10]	0	0	3
MWP-10	[0; 10]	[0;10]	10	2.1257	9.7794×10^{-2}
MWP-11	[0; 10]	[0;10]	7.0441	10	4.2488×10^{-2}
MWP-12	$[-0.5; 0.5]\times[0; 1]$	$[0;10]^2$	0.5	0	0.25
			0.25	0	
MWP-13	$[-1; 3]^2$	$[0;10]^2$	1	Any	1
			1	Any	
GFf-1	$[-5.14; 5.14]^{n_D}$	$[-5.14; 5.14]^{n_U}$	0	$\pm$ 4.5230	$10(n_D + n_U) - 10n_D +$
			...	...	$+30.3533n_U - 5$
			0	$\pm$ 4.5230	
	$n_D = 2$	$n_U = 2$			75.7066
GFf-2	$[0; 10]^{n_D}$	$[0;10]^{n_U}$	$[5]^{n_D}$	$[5]^{n_U}$	0

is $n_{\text{pop}} = 2$ and the convergence threshold of DE is $\rho = 0.25$.

5.5 Results

The results are presented and explained in this section. In particular, four sets of tests have been performed. In the first, Algorithm 2 has been combined with the optimiser *fmincon*, while in the other cases MP-AIDEA has been used. First, we consider one unimodal problem. The performance of the algorithm is assessed increasing the dimension of the problem. Then we consider the worst-case complexity analysis on the benchmark presented in Section 5.4.1 with a wide variety of difficulties. A complexity analysis of the algorithm convergence is then presented for a selected test case for different problem dimensions. Finally we apply Algorithm 2 to solve a real engineering problem: the design for robustness of a communication satellite.

Table 5.4: Reference solutions for the test cases in Table 5.1 with the constraint changing the global optimum

ID	n_D	n_U	d min-max	u min-max	f min-max
GFf-2 & GFc-8	10	10	$[4]^{10}$	$[6.1712]^{10}$	1.4261
	20	20	$[4]^{20}$	$[6.1712]^{20}$	2.8522
	30	30	$[4]^{30}$	$[6.1712]^{30}$	4.2784
	40	40	$[4]^{40}$	$[6.1712]^{40}$	5.7045
	50	50	$[4]^{50}$	$[6.1712]^{50}$	7.1306

Algorithm 10 Success Rate

Define the parameters for algorithm A to solve the constrained problem CP;
Define the number of repetition n;
Define tolerances tol_f, tol_d and tol_u;
Initialise the index of performance $j_s = 0$;
for k = [1,2,.., n] **do**
  Run A on CP with the defined settings;
  Compute $f^k_{\text{opt}} \leftarrow A(CP(\mathbf{d}, \mathbf{u}))$;
  Compute $c^k_{\text{opt}} \leftarrow A(CP(\mathbf{d}, \mathbf{u}))$;
  Compute $\mathbf{d}^k_{\text{opt}}$: optimal solution for the design vector
  Compute $\mathbf{u}^k_{\text{opt}}$: optimal solution for the uncertain vector
  Compute $\delta^k_f = |f_{ref} - f^k_{\text{opt}}|$;
  Compute $\delta^k_d = ||\mathbf{d}_{ref} - \mathbf{d}^k_{\text{opt}}||$;
  Compute $\delta^k_u = ||\mathbf{u}_{ref} - \mathbf{u}^k_{\text{opt}}||$;
  Compute $\delta^k_c = \max(0, c^k_{\text{opt}})$;
  if $\delta^k_c \leq 0 \wedge \delta^k_f < tol_f \wedge \delta^k_d < tol_d \wedge \delta^k_u < tol_u$ **then**
    $j_s = j_s + 1$
  end if
end for
$SR = \frac{j_s}{n}$

5.5.1 Uni-Modal Test Problem

For the first set of results, the test case used is given by the combination of the objective function *GFf-2* and the constraint function *GFc-8*. They are both continuous, differentiable, unimodal and non-separable. With these features a local optimiser is sufficient to solve Eqs. (5.3) to (5.5) at each iteration. The constraint function c admits only one feasible design vector, which is different from the unconstrained optimum of *GFf-2*. The local optimiser we used in this test is *fmincon*. The test functions are devised to be scalable with a predictable value of the exact min-max solution. Results are collected in Table 5.5 for a number of function evaluations up to $7e6$. The table shows up to dimension $n_D = 40$ and $n_U = 40$ the algorithm can achieve $SR = 1$ within the maximum number of function evaluations. For $n_D = 50$ and $n_U = 50$, $7e6$ is not enough and the best result is a success rate of 30%.

Table 5.5: Success Rates of *GFf-2* and *GFc-8* for different problem dimensions (rows) and limits on the maximum number of function evaluations (columns). Optimiser: fmincon. $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

dim	2e5	4e5	6e5	8e5	1e6	2e6	3e6	4e6	5e6	6e6	7e6
10 × 10	0.03	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
20 × 20	-	-	-	-	0.28	1.00	1.00	1.00	1.00	1.00	1.00
30 × 30	-	-	-	-	-	0.14	0.92	1.00	1.00	1.00	1.00
40 × 40	-	-	-	-	-	0.7	0.8	0.34	0.79	0.97	1.00
50 × 50	-	-	-	-	-	-	-	-	0.04	0.16	0.30

5.5.2 Multi-Modal Test Problems

For the second set of experiments, Tables A.3, A.5, A.7, A.9, A.11, A.13 and A.15 collect the results for all the test cases given by the combination of objective functions f from Table 5.1 and constraint functions c from Table 5.2. The last two columns of each table, $n_{\text{iter,min}}$ and $n_{\text{iter,max}}$, collect the minimum and maximum number of loops for which the algorithm achieves SR=1 (rows with the symbol $-$ correspond to problems for which SR=1 has not been obtained for any of the 100 runs). For almost all the problems Algorithm 2 converges to the correct solution with an SR=1 within the maximum number of function evaluations. For some problems (namely *GFF1-GFC1*, *MWP10-GFC4*, *MWP11-GFC5*, *MWP11-GFC2*) few of the runs did not converge to the correct minimum of f but the SR is still reasonably high.

5.5.3 Convergence Complexity

Sections 5.5.1 and 5.5.2 show the performance of Algorithm 2 with respect to the worst-case computational complexity where an upper bound on the number of function evaluations ($n_{\text{feval,max}}$) is fixed for each optimisation step of the approach: $n^{\text{outer}}_{\text{feval,max}}$, $n^{\text{inner,f}}_{\text{feval,max}}$ and $n^{\text{inner,c}}_{\text{feval,max}}$. It is here interesting to show an other computational complexity analysis that is related to the order of magnitude of the number of function evaluations needed to converge to the optimal solution. In particular, Table 5.6 summarises the results for the test case GFf-1&GFc-1. This test case has been selected as representative because it is scalable in both design D and uncertain U spaces and it is also one of the most difficult within the proposed benchmark. Each row in Table 5.6 corresponds to a different problem dimension: $n = n_D = n_U = 2, 3, ..., 10$. The columns represent the average costs at convergence over 10 repetitions for the constrained minimisation in the outer loop in Eq. (5.3) ($\overline{n}^{out,f}_{feval}$), the constrained maximisation in the inner loop in Eq. (5.4) ($\overline{n}^{in,f}_{feval}$), the maximisation of the constraint function in the inner loop in Eq. (5.3) ($\overline{n}^{in,c}_{feval}$) and finally the average cost for the whole algorithm ($\overline{n}_{feval}$). The quantities $\overline{n}^{out,f}_{feval}$, $\overline{n}^{in,f}_{feval}$ and $\overline{n}^{in,c}_{feval}$ have been determined averaging the sum, for the different algorithm's iterations, of the number of function evaluations at convergence. The optimiser MP-AIDEA has been used. In order to assure convergence in each optimisation step in Algorithm 2, the number of

populations has been set equal to the problem dimension $n_{pop} = n$ (the problem is highly multi-modal) and the maximum allowed number of function evaluations of each step has been fixed at $n^{\text{outer}}_{\text{feval,max}} = n^{\text{inner,f}}_{\text{feval,max}} = n^{\text{inner,c}}_{\text{feval,max}} = 1e4n$ while for the whole algorithm it is $n_{\text{feval,max}} = 2e6n$. The remaining input parameters for MP-AIDEA have been fixed as in Section 5.4.3.

Table 5.6: Problem GFf-1&GFc-1. Convergence complexity

dim	$\overline{n}^{out,f}_{feval}$	$\overline{n}^{in,f}_{feval}$	$\overline{n}^{in,c}_{feval}$	$\overline{n}_{feval}$
2	667.1	493.5	162.5	1323.1
3	2631.8	1232.2	200.5	4064.5
4	9607.1	2229.9	241.0	12078.0
5	17405.2	3151.2	254.5	20810.9
6	74085.8	13971.8	645.0	88702.6
7	86616.2	16354.2	992.6	103963.0
8	184190.9	29084.7	1337.4	214613.0
9	211740.5	43107.5	1591.9	256439.9
10	234045.3	51374.4	1765.1	287184.8

5.6 Application: Robust Space System Design

The min-max approach in Algorithm 2 is finally tested on the design for robustness of a CEdS under uncertainty. The system under analysis is an observation spacecraft and the goal of the mission is the fire detection within a belt centred at the latitude of 50 deg. The spacecraft is modelled as the network shown in Fig. 5.12 where the nodes correspond to its subsystems and the links to the coupling between them. The mathematical models that have been used for the nodes are a modification of the ones the authors extensively presented in [13]. The differences are described in the following and are in the explicit definition of a node for the orbital dynamic and in the payload subsystem. Design and uncertain variables are listed in Tables 5.7 and 5.8 respectively.

Within the orbit node, considering a circular Low Earth Orbit (LEO), the altitude h, inclination i, the minimum elevation angle $\epsilon_{\min}$ at which the ground station is able to see the orbiting satellite and the magnetic latitude lat_m are used to evaluate the coupling variables with the Attitude and Orbit Control System (AOCS), the TTC, the Power System and the Payload System (Fig. 5.12). These coupling variables are: the period of each orbit P_o, the number of orbits N_o the satellite perform due to the shift of the longitude of the ascending node, the time of eclipse for each orbit T_{ecl}, the dynamic pressure p_{dyn}, the mean Earth magnetic field strength K_m, the gravitational field K_g, the maximum distance to the target $D_{\max}$, and the access time T_{ac} (or total time in view) between the target and the satellite, where the target is the ground station at 22 deg of latitude used for down-link and up-link. With exception of T_{ac} the formulas can be found in [55]. Instead, considering that the satellite ground track is determined by the inclination i and by the longitude of the ascending node L_{node} and that the latter

increases by 360 *deg* in 1346 *min* (the rotation of the Earth relative to the stars), T_{ac} is calculated considering the total number of orbits i that happens during this period $T_{\text{ac}} = \sum_i T_{\text{ac},i}$. Following [31], that describes the motion of the satellite as seen from a point on the Earth (the ground station), $T_{\text{ac},i}$ is evaluated as:

$$T_{\text{ac},i} = \frac{P}{180deg} \arccos \frac{\cos \lambda_{\text{max},i}}{\cos \lambda_{\text{min},i}} \tag{5.26}$$

with $\lambda_{\text{min},i}$ and $\lambda_{\text{max},i}$ the minimum and maximum Earth Central Angle for the i-th orbit.

The Payload System is an infrared camera that is used to detect possible fires and its target is the belt at 50 *deg* of latitude. Within the payload node the model's parameters are h (shared with the orbit node), P_o (coupling parameter), the width for square detector d, the quality factor Q, the operating wavelength λ, the maximum incidence angle of the instrument IA_{max} and the maximum ground sampling distance Y_{max}. The model evaluates the following coupling variables: the data volume DV shared with OBDH and the power requirement P shared with the Power System. The model evaluates also the payload mass and the percentage of coverage area PC of each orbit during which the payload target is seen. In particular, PC is calculated following [31] as function of λ_{max}, i and the latitude of the target $Lat = 50$ *deg*:

$$PC = \begin{cases} 0 & \text{if } Lat > \lambda_{\text{max}} + i \\ \phi_1/180 & \text{if } i + \lambda_{\text{max}} > Lat > i - \lambda_{\text{max}} \\ (\phi_1 - \phi_2)/180 & \text{if } i - \lambda_{\text{max}} > Lat > 0 \end{cases} \tag{5.27}$$

where

$$\cos \phi_1 = \frac{-\sin \lambda_{\text{max}} + \cos i \sin Lat}{\sin i \cos Lat} \tag{5.28}$$

and

$$\cos \phi_2 = \frac{\sin \lambda_{\text{max}} + \cos i \sin Lat}{\sin i \cos Lat} \tag{5.29}$$

The remaining couplings between nodes are the compressed data volume DV^c that OBDH send to TTC for down-link to the ground station and the power requirements P of all the nodes (orbit excluded) that the Power sub-system has to make available. Finally, the global outputs of the network are the overall mass M of the satellite, sum of the masses of the components, and the percent coverage PC of payload target land. In the optimisation framework, M is considered to be the performance indicator while PC is the constraint to be satisfied. This mission design problem is translated into the following constrained min-max problem:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} M(\mathbf{d}, \mathbf{u}) \\ &s.t. \\ &PC(\mathbf{d}, \mathbf{u}) \geq \nu \quad \forall \mathbf{u} \in U \end{aligned} \tag{5.30}$$

In order to explore the conflict between f and c, the corresponding Pareto front has been reconstructed. We want to apply here the main min-max method presented

in Algorithm 2. The algorithm has then been repeated using 30 different values for the threshold ν through an ECS approach.

The results of Eq. (5.30) are shown in Fig. 5.13 for which the optimiser MP-AIDEA has been used with the setting specified in Section 5.4.3. In particular, Fig. 5.13(a) presents the Pareto front reconstructed for the different values of ν, while the shape of the front can be understood looking at Fig. 5.13(b,c,d). There are indeed two different geographical targets on the Earth for the defined mission: the ground station (22 *deg* of latitude) used for up-link and down-link by TTC and the area that has to be monitored by the payload for possible fire detection (50 *deg* of latitude).

These two targets are quantified in the respectively node's models by T_{ac} and PC, the latter being the constraint function c and the former having an high impact on the final mass M of the overall spacecraft that is the objective function f. The most influential design parameters with regard to PC and T_{ac} in the trade-off within the set of optimal Pareto points are the altitude $h = d_1$ and the inclination $i = d_3$. Fig. 5.13(b) shows their optimal values while moving in the front, while Fig. 5.13(c,d) show finally the corresponding values of T_{ac} and PC. For low values of ν in the constraint function (left side of the Pareto front) the design solution selects the orbit inclination that maximise the amount of time for the link between the spacecraft' antenna and ground station. This configuration reduces the overall mass M at the expense of the capacity of detect fires (PC). As ν increases, the solutions becomes sub-optimal for Fig. 5.13(c) while maximises the area in Fig. 5.13(d).

Table 5.7: Spacecraft model - design parameters

design parameter	symbol	units	id	LB	UB	sub-system
altitude	h	km	d_1	1000	1400	orbit
min elevation angle ground station	ϵ_{min}	deg	d_2	15	20	orbit
inclination	i	deg	d_3	0	90	orbit
width for square detector	d	μm	d_4	20	40	payload
quality factor for imaging	Q	-	d_5	0.5	2	payload
operating wavelength	λ	μm	d_6	3	6	payload
obdh type	τ_{obdh}	-	d_7	0	1	obdh
compression factor	C	-	d_8	0.2	0.6	obdh
slew angle	sl	deg	d_9	10	60	aocs
time for slew manuvers	t_{sl}	s	d_{10}	10	20	aocs
frequency	f	GHz	d_{11}	7	10	ttc
modulation	β	-	d_{12}	0	1	ttc
amplifier type	τ_{amp}	-	d_{13}	0	1	ttc
cell type	τ_{cell}	-	d_{14}	0	1	power
bus voltage	V_{bus}	V	d_{15}	3	5	power
allowed bus drop	V_{drop}	V	d_{16}	1	3	power

Table 5.8: Spacecraft model - uncertain parameters

uncertain parameter	symbol	units	id	LB	UB	sub-system
magnetic latitude	lat_{m}	deg	u_1	0	10	orbit
maximum incidence angle	$IA_{\max}$	deg	u_2	60	80	payload
max ground sampling distance	$Y_{\max}$	m	u_3	60	80	payload
Δ mass	Δ_{m}	%	u_4	0	20	obdh
Δ power	Δ_{p}	%	u_5	0	20	obdh
antenna efficincy	η_{ant}	-	u_6	0.6	0.9	ttc
antenna gain	G_{ant}	dB	u_7	1	5	ttc
mass distribution network	m_{rfdn}	kg	u_8	0.1	0.5	ttc
cell packing efficiency	η_{pack}	-	u_9	0.8	0.9	power
harness mass factor	k_{harn}	%	u_{10}	1	10	power
worst case angle of incidence	θ	deg	u_{11}	20	40	power
reflectance factor	q	-	u_{12}	0.5	0.7	aocs
residual dipole	m	Am^2	u_{13}	0.0005	0.0015	aocs
delta inertia	δI	-	u_{14}	2	10	aocs

5.7 Conclusion

The chapter has presented a new algorithm for the solution of a class of constrained min-max problems. The class of min-max problems emerges naturally from the need to make robust decisions under uncertainty in the case in which constraints need to be always satisfied. The method is based on the alternation of minimisation and a restoration step. This scheme is fairly optimiser agnostic and we demonstrated its applicability even in the case a simple gradient search is used. For the case in which the min-max solution requires the global exploration of a complex solution space, we have proposed the use of a memetic approach based on Inflationary Differential Evolution. Our complexity analysis has revealed that the algorithm is overall of polynomial complexity with a maximum exponent equal to 2.

The combination of the proposed solution strategy and memetic global optimiser was extensively tested on a new benchmark of objective and constraint functions with a variety of features that can be encountered in real-life applications.

Results show that the algorithm we propose is successful at identifying the constrained min-max solution with a limited number of calls to objective functions and constraints. Such a solution minimises the worst-case realisation of the objective function in the uncertain space while guaranteeing its feasibility in all possible scenarios. The benchmark is complemented by a real case of robust optimisation of space systems.

In the case in which a feasible solution in all the uncertain domains could not be found, we proposed a constraint relaxation procedure to automatically adapt the admissible region.

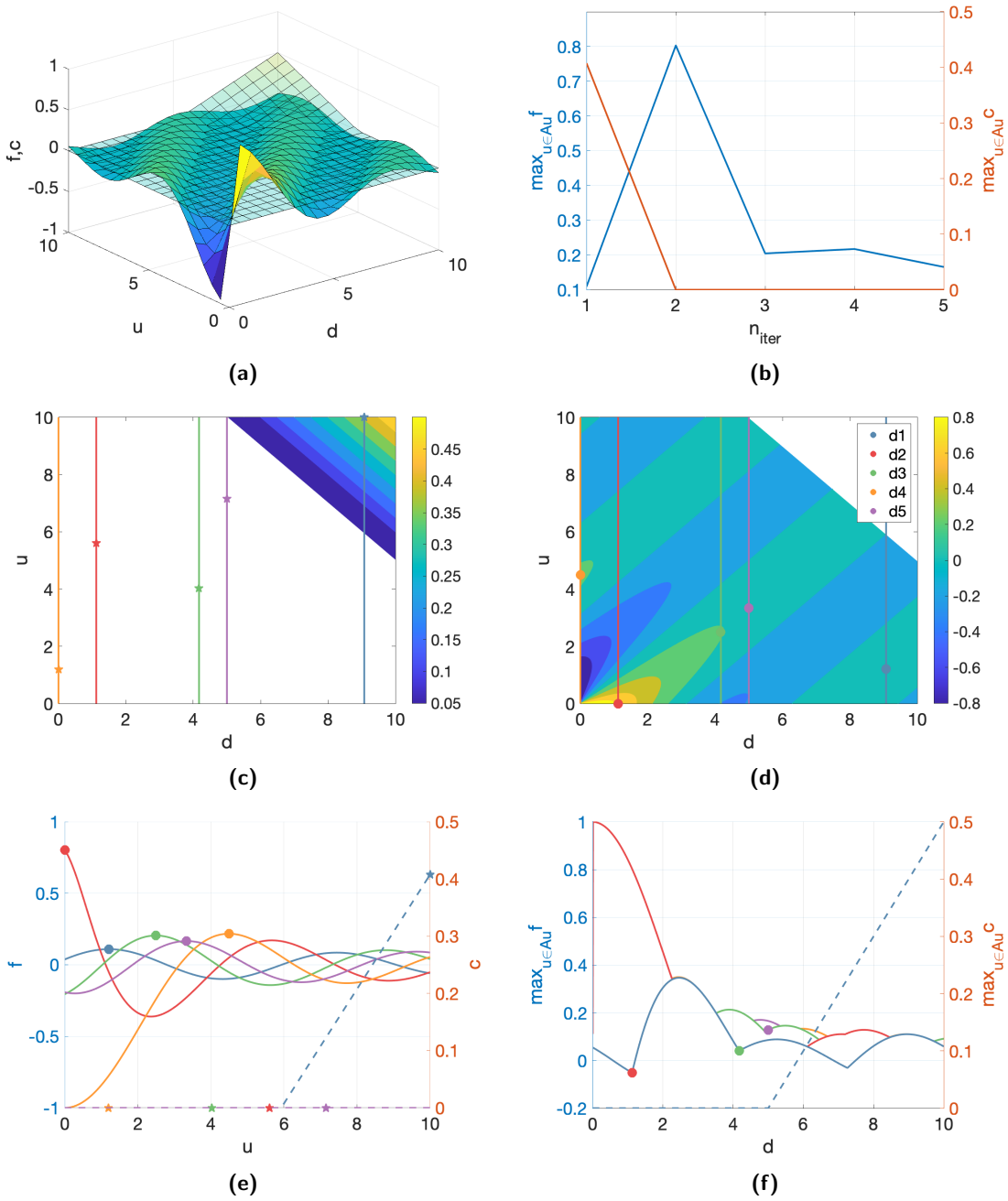


Figure 5.3: Algorithm 2 applied to test case MWP10&GFC1. Sub-figure (a) shows the test case's characteristics by plotting both the objective function f and the constraint function c. Sub-figure (b) shows the convergence of both the worst case conditions $\max f$ and $\max c$ for the design solutions found at each iteration. Sub-figures (c) and (d) plot the functions c and f respectively in the coupled space $D \times U$, all the design solutions at the different iterations (vertical lines) and the corresponding worst case for f and c (dots and stars). The white areas correspond to feasible solutions $c \leq 0$ in (c) and unfeasible solutions f s.t. $c > 0$ in (d). Sub-figure (e) represents, for each explored design configuration, the corresponding f (continuous lines) and c (dotted lines). Sub-figure (f), represents the space of the maxima of f and c over the design space.

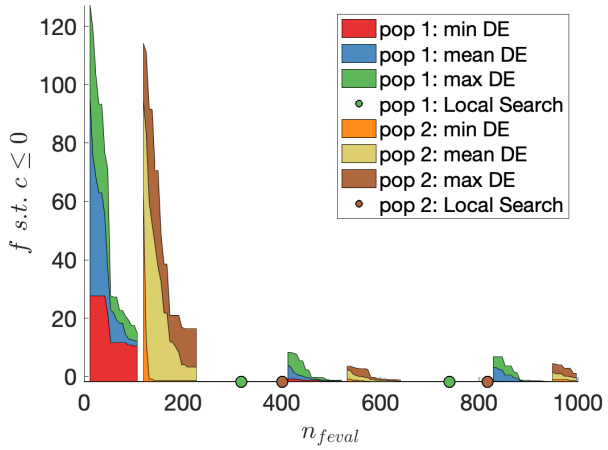
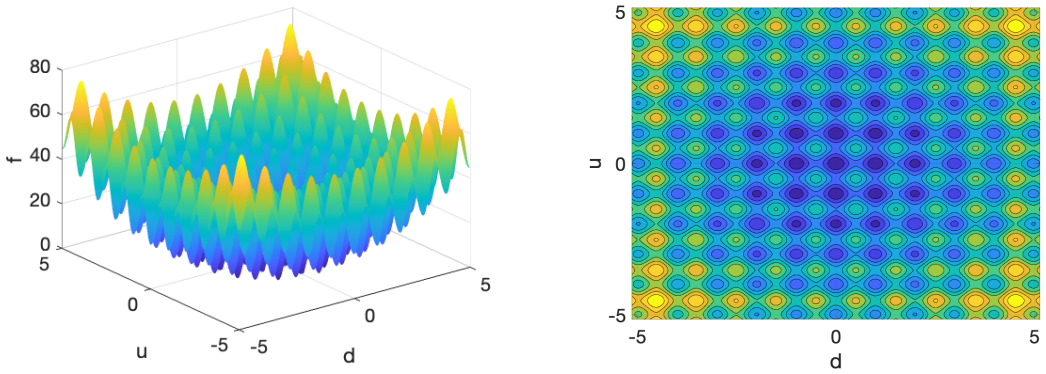
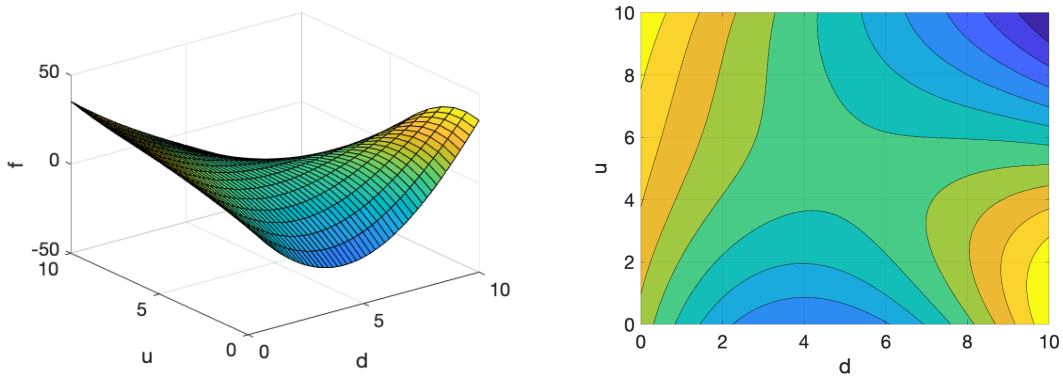


Figure 5.4: Convergence of MP-AIDEA with two populations in the outer loop of Algorithm 2 for test problem *MWP1&GFc1*. Coloured areas represent the convergence of the differential evolution steps (different set of colours for different populations) while dots represents optimal solutions of the local search.



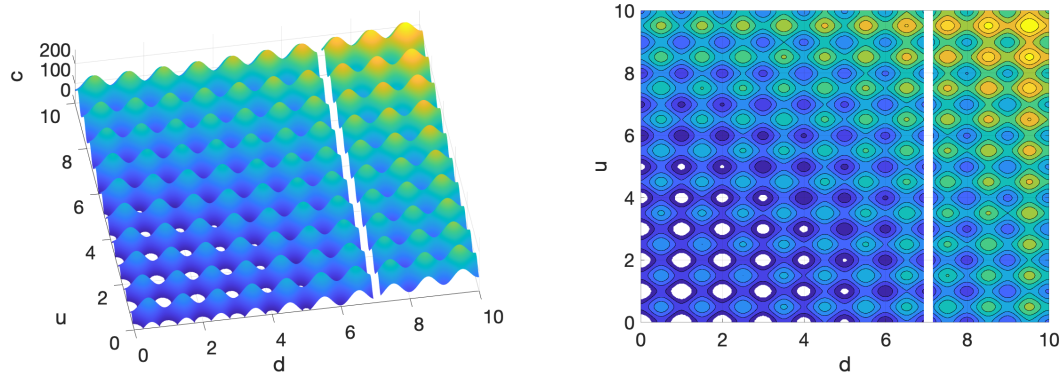
(a) **(b)**

Figure 5.5: GFf-1 in the case $n_D = n_U = 1$



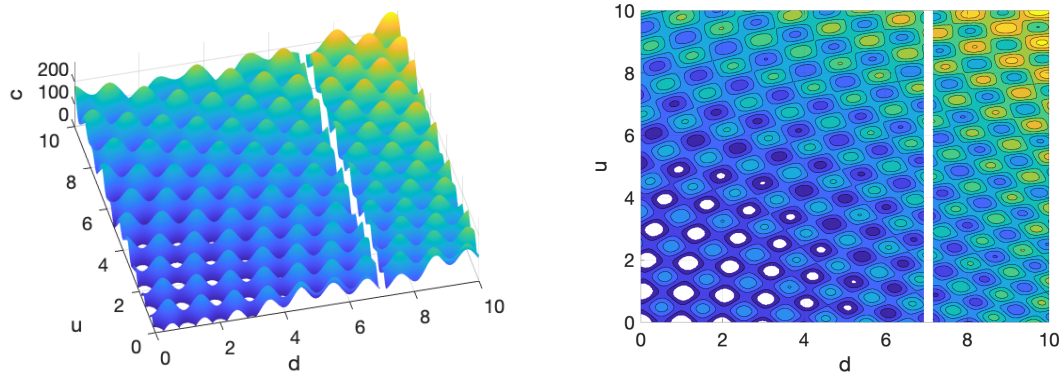
(a) **(b)**

Figure 5.6: GFf-2 in the case $n_D = n_U = 1$



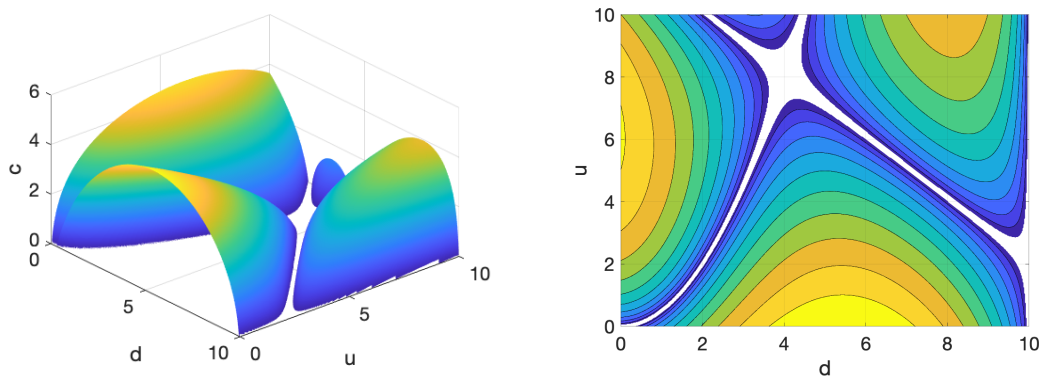
(a) **(b)**

Figure 5.7: GFc-4 applied to MWP-11 in the case $n_D = n_U = 1$. Feasible areas are white.



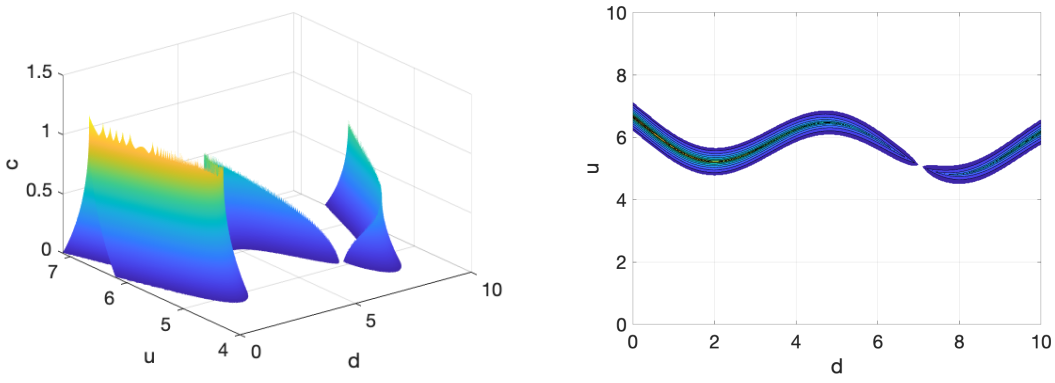
(a) **(b)**

Figure 5.8: GFc-5 applied to MWP-11 in the case $n_D = n_U = 1$. Feasible areas are white.



(a) (b)

Figure 5.9: GFc-6 applied to MWP-10 in the case $n_D = n_U = 1$. Feasible areas are white.



(a) (b)

Figure 5.10: GFc-7am applied to MWP-10 in the case $n_D = n_U = 1$. Feasible areas are white.

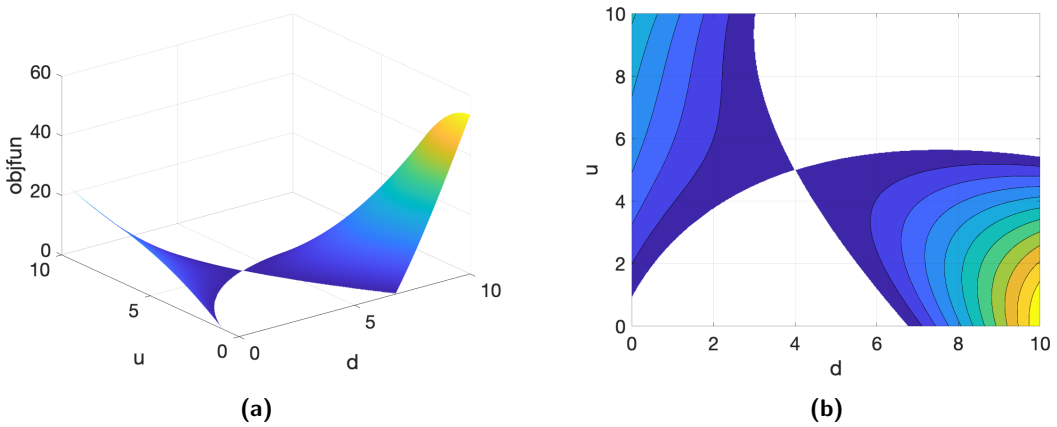


Figure 5.11: GFc-8 applied to MWP-10 in the case $n_D = n_U = 1$. Feasible areas are white.

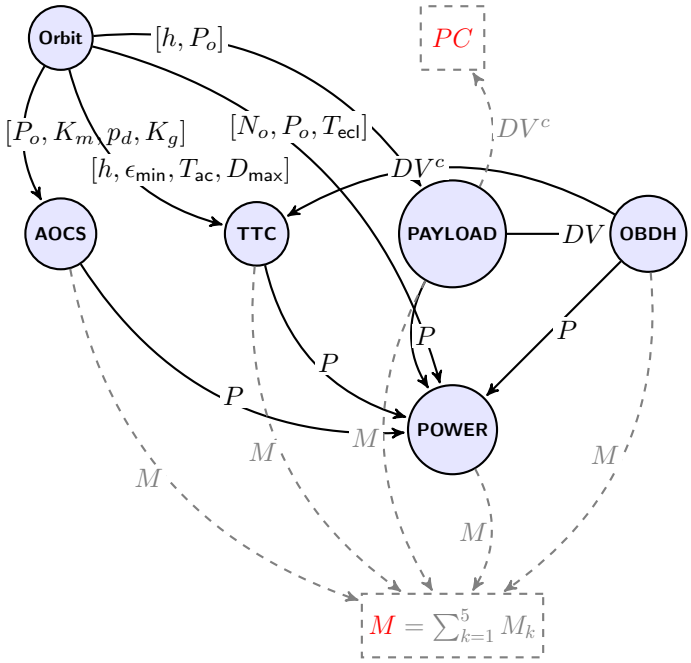


Figure 5.12: Representation of the spacecraft as a complex system. The two quantities of interest are the mass of the M and the percent of coverage are PC for the payload.

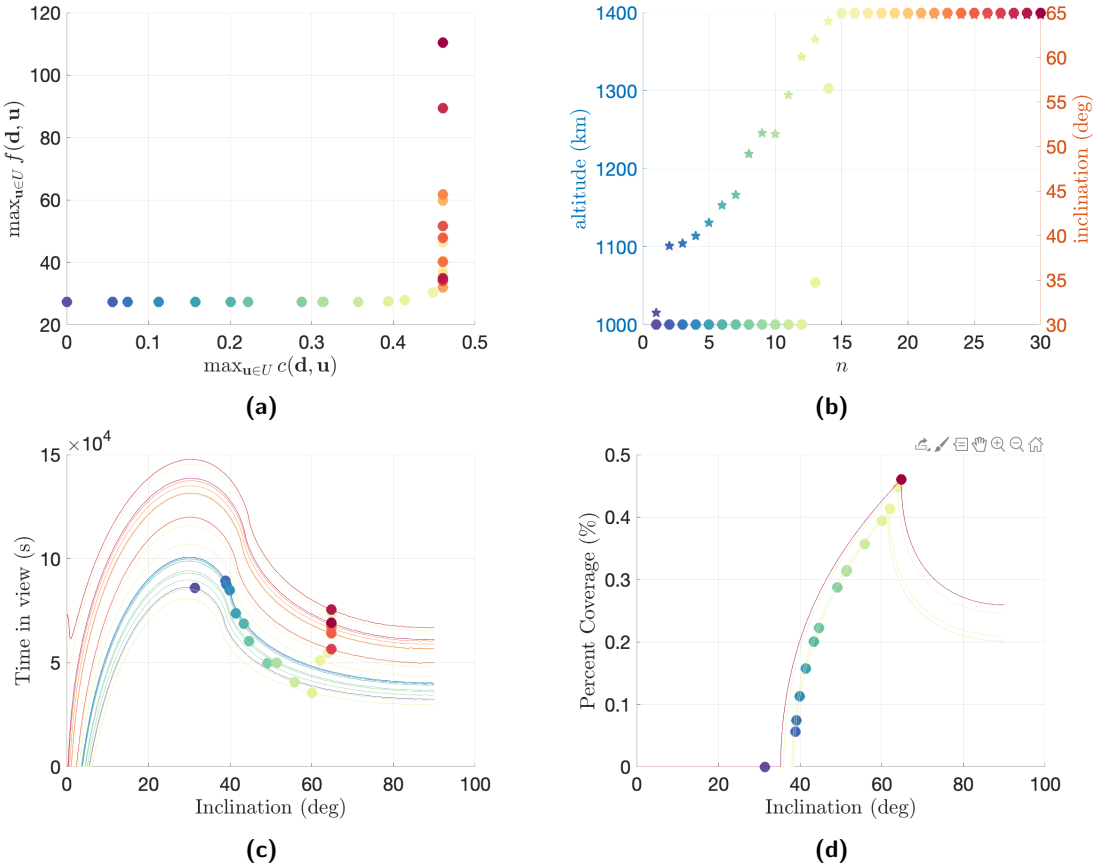


Figure 5.13: Optimal Pareto points for the spacecraft design problem in Eq. (5.30) calculated with Algorithm 2 and applying the EC approach, running 30 optimisations with different thresholds ν in the constraint function c. Sub-figure (a) shows the Pareto Front representing the tension between $\max c$ and $\max f$. Sub-figures (b,c,d) explain the shape of the Pareto Front. The most important design parameters leading the trade-off between f and c are the altitude $H = d_1$ and the inclination $I = d_3$. Sub-figure (b) shows the increase of altitude (points) and inclination (stars) for the different solutions in sub-figure (a). Sub-figure (c) presents the time in view between the satellite and the ground station for a series of revolutions as a function of the inclination. Sub-figure (d) finally shows the per cent of land coverage by the payload.

Part III

System Optimisation Under Epistemic Uncertainty

6

Evidence Network Model for System Design Optimisation

" Remember that all models are wrong;
the practical question is how wrong do they have to be to not be useful. "
– George Box

The content of this chapter was published in:

- **G. Filippi** and M. Vasile, Introduction to Evidence-Based Robust Optimisation, in *Optimization Under Uncertainty with Applications to Aerospace Engineering (M. Vasile, ed.), Springer Nature*, 2020 [58]
- M. Vasile, **G. Filippi**, C. Ortega Absil, and A. Riccardi, Fast belief estimation in evidence network models, in *EUROGEN, (Madrid)*, 9 2017 [61]

The main challenge of MBSE and SDO in the design optimisation of CEdS is related to the increasing complexity of the system under design. Inter-connectivity between sub-systems and uncertainty forces the engineer modeller to abandon reductionism and embrace instead a holistic approach. Modelling these interactions and couplings is indeed of fundamental importance, much more than pushing the fidelity of the single components' models.

This chapter presents a modelling framework to capture CEdgS in the presence of uncertainty and within the framework of SDO for MBSE. The developed approach is called ENM: a holistic network modelling procedure which captures the interdependence and the sharing of information between sub-systems and also can be used to quantify epistemic uncertainty. As the name suggests, DST of Evidence is adopted to quantify uncertainty as explained in the following chapter. The ENM was first introduced in [193]. The method was extended in [61] to make UQ through ENM computationally more efficient. [62] introduced a time-dependent reliability measure in the ENM and finally [13] introduced the concept of resilience in a higher level optimisation framework.

The necessary background on system modelling has been presented in Chapter 2. The methodology will be further extended in the following chapters for uncertainty quantification and optimisation under uncertainty with DST.

The modelling approach of ENM is presented in Section 6.1. A simple example of the network model is then given in Section 6.2. Section 6.3 define our problem formulation with regards to SDO under epistemic uncertainty. Conclusions of the chapter are finally presented in Section 6.4.

6.1 Evidence Network Model

Consider a generic complex system with N sub-systems and suppose a single QoI has been defined for which we give a mathematical formulation $f : \Omega \to \mathcal{R}$. For each of its sub-systems, a model g_i can be generated that describes the contribution of the $i-$th subsystem to the global metric f. The models in general accept as inputs a vector of decision (or design) variables $\mathbf{d}$ and a vector of uncertain (or environmental) variables $\mathbf{u}$. The proposed modelling approach is called ENM and translates the complex system into a non-directed graph (or network) where the corresponding N sub-systems' models are the nodes of the graph. The topology of the network represents the sub-system inter-relations and the exchange functions h are used to quantify them. The ENM of the generic metric f can then be represented as:

$$f(\mathbf{d}, \mathbf{u}) = \sum_{i=1}^{N} g_i(\mathbf{d}, \mathbf{u}_i, \mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})) \tag{6.1}$$

In Eq. (6.1) each node i is characterised by the value function g_i and exchanges information with node j via the exchange function h_{ij}. Functions f, g and h depends on design $\mathbf{d}$ and uncertain $\mathbf{u}$ vector. $\mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})$ is the vector of scalar functions $h_{ij}(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})$, $j \in J_i$ and J_i is the set of indexes of nodes connected to the i-th node; $\mathbf{u}_i$ are the uncertain variables of subsystem i not affecting any other subsystems and $\mathbf{u}_{ij}$ are the uncertain variables affecting both subsystems i and j. Note that accordingly to this notation $\mathbf{u}_{ij} = \mathbf{u}_{ji}$. We then call $\mathbf{u}_i$ *uncoupled* variables because they influence only subsystem i and $\mathbf{u}_{ij}$ *coupled* variables because they influence subsystems i and j. If the same parameters are shared between nodes i, j and k, it is: $\mathbf{u}_{ij} = \mathbf{u}_{ik} = \mathbf{u}_{jk}$. It is also important to note that if values of h_{ij} functions were known with certainty the nodes composing the network would be decoupled and statistically independent.

ENM has been developed under some properties as explained in the following. Consider the pair (f, Ω) where $f : \Omega \to \mathcal{R}$ is the global model. Furthermore, we introduce the two sets Ω_x and Ω_y such that $\Omega = \Omega_x \times \Omega_y$. Consider now two partitions D_x and D_y respectively of Ω_x and Ω_y. Given $\delta\Omega_x^p \in D_x$ and $\delta\Omega_y^q \in D_y$ we compute:

$$\mathbf{y}_0 = \arg\max_{\delta\Omega_y^q} f(\mathbf{x}_0, \mathbf{y}) \tag{6.2}$$

for an arbitrary initial $\mathbf{x}_0 \in \Omega_x$ and the iteration:

$$\mathbf{x}_k = \arg\max_{\delta\Omega_x^p} f(\mathbf{x}, \mathbf{y}_{k-1}) \tag{6.3}$$

$$\mathbf{y}_k = \arg\max_{\delta\Omega_y^q} f(\mathbf{x}_k, \mathbf{y}) \tag{6.4}$$

We say that the pair (f, Ω) is M-decomposable if, given an $M \geq 0$, for $k > M$ we have that:

$$(\mathbf{x}_k, \mathbf{y}_k) = \arg\max_{\delta\Omega^{pq}} f(\mathbf{x}, \mathbf{y}) \tag{6.5}$$

with $\delta\Omega^{pq} = \delta\Omega_x^p \times \delta\Omega_y^q$.

The ENM properties are then:

1. The contribution of the coupled variable $\mathbf{u}_{ij}$ to the value f manifests through the scalar functions h_{ij} and h_{ji}.

2. The pair (f, Ω) is M-decomposable. In particular, we consider the case in which $M = 0$.

6.2 Evidence Network Model Example

As an example, a generic system with 3 sub-systems is considered where the global network QoI f is supposed to be the cumulative sum over the subsystems quantities g_i. The system can then be modelled as:

$$\begin{aligned} f(\mathbf{d}, \mathbf{u}) &= f_1(\mathbf{d}, \mathbf{u}) + f_2(\mathbf{d}, \mathbf{u}) + f_3(\mathbf{d}, \mathbf{u}) \\ &= d_1 d_2 + u_1 + u_1 \sin(d_2 u_4) + u_1 \cos(u_1 u_6) + \\ &\quad d_1 + u_2 + u_4 \sin(u_2 + u_4) + u_5 \sin(u_2 u_5) + \\ &\quad \tfrac{d_1}{d_2} + u_3 + u_6 \sin(u_3 + u_6) + u_5 \sin(u_3 u_5). \end{aligned} \tag{6.6}$$

It is possible to classify the uncertain vectors as coupled $\mathbf{u_{ij}}$ and uncoupled $\mathbf{u_i}$: $\mathbf{u_1} = u_1$, $\mathbf{u_2} = u_2$, $\mathbf{u_3} = u_3$, $\mathbf{u_{12}} = u_4$, $\mathbf{u_{23}} = u_5$ and $\mathbf{u_{13}} = u_6$.

The corresponding ENM consists of a fully connected network with 3 nodes as in Fig. 6.1 and in the following formulation:

$$\begin{aligned} f(\mathbf{d}, \mathbf{u}) &= g_1(\mathbf{d}, \mathbf{u_1}, h_{12}(\mathbf{d}, \mathbf{u_1}, \mathbf{u_{12}}), h_{13}(\mathbf{d}, \mathbf{u_1}, \mathbf{u_{13}}))) + \\ &\quad g_2(\mathbf{d}, \mathbf{u_2}, h_{21}(\mathbf{d}, \mathbf{u_2}, \mathbf{u_{12}}), h_{23}(\mathbf{d}, \mathbf{u_2}, \mathbf{u_{23}}))) + \\ &\quad g_3(\mathbf{d}, \mathbf{u_3}, h_{31}(\mathbf{d}, \mathbf{u_3}, \mathbf{u_{13}}), h_{32}(\mathbf{d}, \mathbf{u_3}, \mathbf{u_{23}})). \end{aligned} \tag{6.7}$$

where f is the global network quantity dependant on the full vectors $\mathbf{d}$ and $\mathbf{u}$. The functions g_i with $i = 1, 2, 3$ are the specific functions of sub-systems 1,2,3. Similarly, $\mathbf{u_i}$ is the vector of the uncertain components affecting only subsystem i. Finally $\mathbf{h_{ij}}$ pertain to the coupling vectors $\mathbf{u_{ij}}$ with $i = 1, 2, 3$.

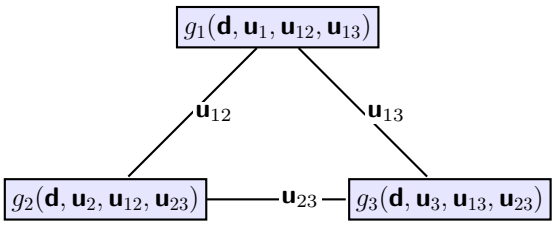


Figure 6.1: Evidence Network Model of a generic system F composed of three sub-systems with coupled variables u_{12}, u_{13} and u_{23}.

6.3 System Design Optimisation Under Epistemic Uncertainty

It here is suggested to extend the general formulation of OUU in Eq. (4.7) with the SDO framework in Eq. (4.20) where UQ is modelled with DST:

$$\begin{array}{ll} \max & \text{Belief}\Big(\sum_{i=1}^{N}\left[f_i\left(\mathbf{d}, \mathbf{u}_i, \mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})\right)\right] \leq \nu\Big) \\ \min_{\nu\in\mathbb{R}} & \nu \\ \text{w.r.t.} & \mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij} \\ \text{s.t.} & \text{Belief}\Big(c_i(\mathbf{d}, \mathbf{u}_i, \mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})) \leq \nu_c\Big) \geq 1-\epsilon \quad \text{for i} = 1, \ldots, \text{N} \end{array} \tag{6.8}$$

Finally, Eq. (6.8) can be restricted to the most conservative condition, given by the minimum ν such that Belief$(f < \nu) = 1$ and by $\epsilon = 0$. Eq. (6.8) then translates into the deterministic bi-level min-max problem:

$$\begin{array}{l} \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d}, \mathbf{u}) \\ s.t. \\ \max_{\mathbf{u}\in U} c(\mathbf{d}, \mathbf{u}) \leq \nu_c. \end{array} \tag{6.9}$$

6.4 Conclusion

This chapter introduced the reader to the ENM framework developed for the modelling of CEdS under epistemic uncertainty. It has been explained how the ENM can be constructed starting from the basic models of the sub-systems. A simple example has been given. The contribution of this framework will be exploited more in the next chapters.

7

Evidence-Based Robust Optimisation

The content of this chapter was published in:

- C. O. Absil, M. Vasile, **G. Filippi**, A. Riccardi, and M. Vasile, A Variance-Based Estimation of the Resilience Indices in the Preliminary Design Optimisation of Engineering Systems Under Epistemic Uncertainty, in *EUROGEN, (Madrid)*, 2017 [60]
- M. Vasile, **G. Filippi**, C. Ortega Absil, and A. Riccardi, Fast belief estimation in evidence network models, in *EUROGEN, (Madrid)*, 9 2017 [61]
- **G. Filippi**, M. Marchi, M. Vasile, and P. Vercesi, Evidence-Based Robust Optimisation of Space Systems with Evidence Network Models, in *IEEE Congress on Evolutionary Computation (CEC), (Rio De Janeiro)*, 2018 [62].
- **G. Filippi**, M. Vasile, P. Z. Korondi, M. Marchi, and C. Poloni, Robust design optimisation of dynamical space systems, in *8th International Systems & Concurrent Engineering for Space Applications Conference*, (Glasgow), 2018 [63].
- **G. Filippi** and M. Vasile, Introduction to Evidence-Based Robust Optimisation, in *Optimization Under Uncertainty with Applications to Aerospace Engineering (M. Vasile, ed.), Springer Nature*, 2020 [58]

In the early phase of the design of an engineering system, there is a degree of uncertainty on its parts and configurations. This uncertainty is often epistemic in nature and translates into an uncertainty in the performance of the system as a whole. With this regard, this chapter explores two problems and proposes algorithmic solution methodologies. The first problem is the definition of an holistic formulation and solution methodology for OUU with epistemic uncertainty in the filed of SDO. The second problem, included in the former, is related to the computational complexity associated

with UQ within SDO under uncertainty and UMDO [79]. This is particularly important when non-probabilistic uncertainty theories, DST as an example, are adopted.

The chapter introduces EBRO, a methodology to account for epistemic uncertainty in the design for robustness of CEdSs. Epistemic uncertainty is modelled with DST which offers a natural way to assign degrees of belief to the expected performance of a system and to rigorously quantify the impact of epistemic uncertainty on the associated QoI. In EBRO the system design is optimised to maximise performance under epistemic uncertainty. The resulting system OUU process is composed of the solution of a constrained bi-level min-max optimisation problem followed by the reconstruction of the Belief and Plausibility on the value of the performance metric. The methodology for the solution of CSOP and CMOP min-max are presented in Chapter 5 and Appendix B.

We also present two novel methodologies for UQ with DST that reduce the complexity time for the evaluation of Belief and Plausibility within the framework of DST and make possible their application to MBSE. The first methodology is an outer approximation via decomposition approach and is here presented in detail. The second methodology is a binary tree approach and is explained in Appendix C. We can identify five main contributions given by the proposed methodologies. (i) They both drastically reduce the computational cost making then accessible the use of DST within the framework of SDO and MBSE in general. This is important because UQ within the framework of MBSE is intractable even for simple problems when no proper computational reduction techniques are implemented. (ii) The decomposition approach presents a compact graph representation of CEdgS. It is customary in MDO to represent a system as a set of connected components that exchange information through connecting links. An example of UMDO with Evidence Theory can be found in [171] and an application of the Design Structure Matrix (DSM) is in [142]. However, in an ENM the specific properties of the nodes and the form in which they exchange information is such that Belief functions can be computed in polynomial time. Within ENM the correlations between nodes are represented by scalar values that model in a compact way the influence of many uncertain parameters and weight the different links. Also ENM allows for an easier representation of sub-networks and clusters. (iii) It is proposed a quantification of the computational complexity. (iv) They are both *outer approximations*. This means that they are conservative iterative procedures that produce estimated values of $Belief$ and $Plausibility$ which are lower than the actual ones. Conservative methods are privileged since it is better to overestimate uncertainty than underestimate it. (v) The methods iteratively converge to the exact solution with a sequence of outer approximations. Further computational time can then be spent to refine the solution if needed.

The framework presented in this chapter finds applicability in the design and optimisation of complex aerospace systems, composed of a number of interconnected components, the behaviour of which can not be inferred only by the behaviour of each of its parts.

The reader can find the necessary background in Chapter 3 for the UQ problems and algorithms. The reader finds an overview of the different sources and forms of uncertainty, explains the difference between *aleatory* and *epistemic* uncertainty and focuses the attention on DST. It also gives an overview of the existing algorithms

for the solution of the UQ problem. Section 4.5 are important for an introduction to respectively the SDO problem and the related solvers in the literature. Chapter 5 is important for the worst case optimisation approach that has been developed. A further developed methodology for which I gave a partial contribution, is instead presented in Appendix C. It presents a technique based on an Evolutionary Binary Tree approach (more details are in [60]). It is explained the method's philosophy, the heuristics implemented and finally in Appendix C.2.1 the method is applied for verification to the preliminary reliability-based design of the solar array of a small spacecraft. Results are then presented and discussed.

Section 7.1 provides the mathematical formulation of the general EBRO problem. Section 7.2 presents the outer approximation via decomposition approach methodology which is based on the ENM procedure. A set of benchmark functions is proposed and used for testing. The results are finally presented and discussed. Section 7.3 presents a engineering application problem. The system's and sub-system's models and the application of the method is described. Section 7.4 finally presents the conclusion.

7.1 Problem Formulation and Solution Methodology

This section explains the EBRO approach used to incorporate epistemic uncertainty in the optimisation process and to design the system for robustness.

In the general formulation, we want to minimise the threshold ν and maximise the belief in the statement $f(\mathbf{d}, \mathbf{u}) \leq \nu$ while maintaining a hard condition on the constraint satisfaction:

$$\begin{aligned} &\max_{\mathbf{d}\in D} Bel(f(\mathbf{d}, \mathbf{u}) \leq \nu) \\ &\min_{\nu\in\mathbb{R}} \nu \\ &Bel(c(\mathbf{d}, \mathbf{u}) \leq \nu_c) > 1 - \epsilon \end{aligned} \tag{7.1}$$

Eq. (7.1) requires the evaluation of the belief curve for both the functions f and c and it becomes easily intractable. In fact, there is a dependence of the belief to the design vector $\mathbf{d}$ and the thresholds ν and ν_c thus for each new value of $\mathbf{d}$, ν and ν_c the belief has to be revalued.

Among all vectors $\mathbf{d}$ that solve problem (7.1) the most critical one, $\mathbf{d}^*$, corresponds to the minimum values of ν and ν_c such that $Bel(f(\mathbf{d}, \mathbf{u}))$ is maximum and $Bel(C(\mathbf{d}, \mathbf{u}) \leq \nu_c)) = 1$. We call the search for $\mathbf{d}^*$, worst-case scenario optimisation and it can be formulated as the deterministic min-max optimisation problem [11]:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d}, \mathbf{u}) \\ &s.t. \\ &\forall \mathbf{u} \in U: \quad c(\mathbf{d}, \mathbf{u}) \leq 0. \end{aligned} \tag{7.2}$$

Solving for the worst-case scenario makes the optimisation problem independent of the uncertainty quantification method, has a complexity that is independent of the number of FEs and does not require any particular assumption on the constraint functions.

The approach then involve the solution of a constrained min-max problem, to deliver a robust design point. Starting from this robust design point a sequence of evolutionary optimisation steps are used to reconstruct an approximation of the Belief and

Plausibility curves associated to a particular design solution.

7.2 Outer Approximation via Decomposition

In some cases the structure and nature of the function f can be exploited to drastically reduce the computation of $Bel(f \leq \nu)$ in (4.13). The following approach is based on the possibility of decomposing the uncertainty space of some UQ problems that happen to be a common situation in SDO. The decomposition method is based on the modelling approach presented in Chapter 6.

The decomposition algorithm aims at decoupling the sub-systems over the uncertain variables in order to optimise only over a small subset of the FEs (Algorithm 11). The procedure requires the following steps:

1. Solution of the optimal worst-case scenario problem:

$$\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d}, \mathbf{u}) \tag{7.3}$$

2. Maximisation over the coupled variables and computation of $Bel_c(A)$.
3. Sample of the partial Belief curves.
4. Maximisation over the uncoupled variables.
5. Reconstruction of the approximation $\widetilde{Bel}(A)$.

Point 1 has been already discussed in Chapter 5. In the following, the solution of problem (7.3) is represented by the values $\tilde{\mathbf{d}}$ and $\underline{\mathbf{u}}$ and it is assumed that $\tilde{\mathbf{d}}$ is already available.

For each coupled vector $\mathbf{u}_{ij}$ a maximisation is run over each FE $\theta_{k,ij} \subseteq \Theta_{ij} \subseteq U$, given $\widetilde{\mathbf{d}}$ and keeping fixed all the other components to $\underline{\mathbf{u}}_{\mathbf{k}}$ and $\underline{\mathbf{u}}_{\mathbf{lm}}$ $\forall k, l, m \setminus \{l, m\} = \{i, j\}$. Taking again the example in Figure 6.1 we have:

$$\begin{aligned}
\hat{\mathbf{u}}_{k,12} &= \arg\max_{\mathbf{u}_{12}\in\theta_{k,12}} f(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_{\mathbf{1}}, \underline{\mathbf{u}}_{\mathbf{2}}, \underline{\mathbf{u}}_{\mathbf{3}}, \mathbf{u}_{\mathbf{12}}, \underline{\mathbf{u}}_{\mathbf{13}}, \underline{\mathbf{u}}_{\mathbf{23}}), \forall\theta_{k,12} \subset \Theta_{12} \\
\hat{\mathbf{u}}_{k,13} &= \arg\max_{\mathbf{u}_{13}\in\theta_{k,13}} f(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_{\mathbf{1}}, \underline{\mathbf{u}}_{\mathbf{2}}, \underline{\mathbf{u}}_{\mathbf{3}}, \underline{\mathbf{u}}_{\mathbf{12}}, \mathbf{u}_{\mathbf{13}}, \underline{\mathbf{u}}_{\mathbf{23}}), \forall\theta_{k,13} \subset \Theta_{13} \\
\hat{\mathbf{u}}_{k,23} &= \arg\max_{\mathbf{u}_{23}\in\theta_{k,23}} f(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_{\mathbf{1}}, \underline{\mathbf{u}}_{\mathbf{2}}, \underline{\mathbf{u}}_{\mathbf{3}}, \underline{\mathbf{u}}_{\mathbf{12}}, \underline{\mathbf{u}}_{\mathbf{13}}, \mathbf{u}_{\mathbf{23}}), \forall\theta_{k,23} \subset \Theta_{23}
\end{aligned} \tag{7.4}$$

For easiness in the notation we will indicate with

$$f(\mathbf{u}_{ij}) := f(\tilde{\mathbf{d}}, \underline{\mathbf{u}}_1, ..., \mathbf{u}_{ij}, ..., \underline{\mathbf{u}}_{i+1j}, ...).$$

We can then compute the partial belief associated only to the coupled variables with index ij:

$$Bel(f(\mathbf{u}_{ij}) < \nu) = \sum_{\theta_{k,ij} | \max_{\mathbf{u}_{ij}\in\theta_{k,ij}} F(\mathbf{u}_{ij}) \leq \nu} bpa(\theta_{k,ij}) \tag{7.5}$$

The calculation of the partial belief can be found in Algorithm 11, line 6. Once the partial belief curve, for each coupled vector, is available, one can sample these curves, by taking a succession of $\{\nu_1, ..., \nu^q, ..., \nu_{N_S} = \nu\}$ values, and find the corresponding values of the coupled vectors $\widehat{\mathbf{u}}^q_{k,ij}$. These values will be used in the next step to decouple the functions g_i (g_j) and compute the maxima of each g_i (g_j) with respect to the uncoupled variables $\mathbf{u}_i$ ($\mathbf{u}_j$).

For each level q, given a fixed value of the coupling functions, one can study each g_i independently of the others. The idea is to run an optimisation for each function g_i over only the uncoupled vector $\mathbf{u}_i$. With the example in Figure 6.1 in mind, having

$$\hat{h}^q_{ij}(\mathbf{u}_i) := h_{ij}(\tilde{\mathbf{d}}, \mathbf{u}_i, \hat{\mathbf{u}}^q_{ij}) \tag{7.6}$$

where $\hat{\mathbf{u}}^q_{ij} := \hat{\mathbf{u}}^q_{k^*,ij} : k^* = \arg\max_k f(\hat{\mathbf{u}}^q_{k,ij})$, is one of the maxima attained by the coupled variable $\mathbf{u}_{ij}$. For every FE $\theta_{k,i} \in \Theta_i$ we have:

$$\begin{aligned}
\hat{\mathbf{u}}^q_{k_1,1} &= \arg\max_{\mathbf{u}_1 \in \theta_{k,1}} g_1(\tilde{\mathbf{d}}, \mathbf{u}_1, \widehat{h}^q_{12}(\mathbf{u}_1), \widehat{h}^q_{13}(\mathbf{u}_1)), \forall \theta_{k_1,1} \subset \Theta_1 \\
\hat{\mathbf{u}}^q_{k_2,2} &= \arg\max_{\mathbf{u}_2 \in \theta_{k,2}} g_2(\tilde{\mathbf{d}}, \mathbf{u}_2, \widehat{h}^q_{21}(\mathbf{u}_2), \widehat{h}^q_{23}(\mathbf{u}_2)), \forall \theta_{k_2,2} \subset \Theta_2 \\
\hat{\mathbf{u}}^q_{k_3,3} &= \arg\max_{\mathbf{u}_3 \in \theta_{k,3}} g_3(\tilde{\mathbf{d}}, \mathbf{u}_3, \widehat{h}^q_{31}(\mathbf{u}_3), \widehat{h}^q_{32}(\mathbf{u}_3)), \forall \theta_{k_3,3} \subset \Theta_3
\end{aligned} \tag{7.7}$$

with the corresponding values $\hat{g}^q_{k_1,1}$, $\hat{g}^q_{k_2,2}$ and $\hat{g}^q_{k_3,3}$.

Once all the maxima over the FEs of the uncoupled variables are available for each sample q one can calculate an approximation of $Bel(f(\mathbf{d}, \mathbf{u}) < \nu)$ as follows.

From Eq. (7.7), for each sample q the maximum associated to the FE $\theta_k = \theta_{k_1,1} \times \theta_{k_2,2} \times \theta_{k_3,3}$, for $k = 1, ...N_{FE,1} \cdot N_{FE,2} \cdot N_{FE,3}$, given the condition of positive semidefinition of g_i, is:

$$\max_{(\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3) \in \theta_k} f(\tilde{\mathbf{d}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \hat{\mathbf{u}}^q_{12}, \hat{\mathbf{u}}^q_{13}, \hat{\mathbf{u}}^q_{23}) = \hat{g}^q_{k_1,1} + \hat{g}^q_{k_2,2} + \hat{g}^q_{k_3,3} \tag{7.8}$$

with associated basic probability assignment:

$$bpa^q(\theta_k) = bpa(\theta_{k_1,1}) bpa(\theta_{k_2,2}) bpa(\theta_{k_3,3}) \Delta Bel^q \tag{7.9}$$

where $\Delta Bel^q = \prod_{ij} \Delta Bel^q_{ij}$ are the contributions of the partial belief curves in (7.5): the generic ΔBel^q_{ij} is the difference of belief between sample q and $q-1$ in the partial belief curve Bel_c about the coupled uncertain vector $\mathbf{u_{ij}}$. In other words, the bpa of each θ_k is the product of all the bpa's of the FE of each uncoupled variable scaled with the product of the belief values of the samples drawn from the partial belief curves (Line 18). The approximation of the belief is then computed as:

$$\widetilde{Bel}(f(\mathbf{d}, \mathbf{u}) \leq \nu) = \sum_q \sum_k bpa^q(\theta_k) \tag{7.10}$$

7.2.1 Complexity Analysis

The number of optimisations for the exact curve reconstruction (equal to the total number of FE) for a problem with m uncertain variables, each defined over N_k intervals, is:

$$N_{FE} = \prod_{k=1}^{m} N_k. \tag{7.11}$$

In terms of coupled and uncoupled uncertain vectors we can write:

$$N_{FE} = \left(\prod_{i=1}^{m_u} \prod_{k=1}^{p_i^u} N_{i,k}^u \right) \left(\prod_{i=1}^{m_c} \prod_{k=1}^{p_i^c} N_{i,k}^c \right) \tag{7.12}$$

where p_i^u and p_i^c are the number of components of the $i-th$ uncoupled and coupled vector, respectively, and $N_{i,k}^u$ and $N_{i,k}^c$ are the number of intervals of the $k-th$ components of the $i-th$ uncoupled and coupled vector respectively. The total number of FE that needs to be explored in the decomposition is instead:

$$N_{FE}^{Dec} = N_s \sum_{i=1}^{m_u} N_{FE,i}^u + \sum_{i=1}^{m_c} N_{FE,i}^c \tag{7.13}$$

considering the vector of uncertainties ordered as

$$\mathbf{u} = [\underbrace{\mathbf{u}_1, ..., \mathbf{u}_{m_u}}_{\text{uncoupled}}, \underbrace{\mathbf{u}_1, ..., \mathbf{u}_{m_c}}_{\text{coupled}}]$$

where N_s is the number of samples of the partial belief curves, $N_{FE,i}^c = \prod_{k=1}^{p_i^c} N_{i,k}^c$ and $N_{FE,i}^u = \prod_{k=1}^{p_i^u} N_{i,k}^u$. This means that the computational complexity to calculate the maxima of the function F within the FEs is polynomial with the number of subsystems and remains exponential for each individual uncoupled or coupled vector.

7.2.2 Method advantages

This section better clarifies with an example the advantages of the proposed method stated in the introduction of this chapter.

Consider the equation $f = g_1 + g_2$ where

$$g_1(\mathbf{d}, \mathbf{u}) = 10u_1^2 + |u_2|u_5^2 + \frac{u_6^4}{100} + d_1|d_2| \tag{7.14}$$

$$g_2(\mathbf{d}, \mathbf{u}) = |u_3| + u_4^2 \frac{|u_5|}{10} + u_6^2 + |d_1| \tag{7.15}$$

with two uncoupled vectors ($\mathbf{u_1} = (u_1, u_2)$ and $\mathbf{u_2} = (u_3, u_4)$) and one coupled vector ($\mathbf{u_{12}} = (u_5, u_6)$). Consider also each uncertain variable to be defined over three uncertain intervals, giving as result a total of $3^6 = 729$ FE.

Fig. 7.1 shows the exact Belief curve together with the approximated one calculated with an increasing number of samples (parameter of the process).

Algorithm 11 Decomposition

Initialise
Uncoupled vectors $\mathbf{u}_u = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_i, ..., \mathbf{u}_{m_u}]$
Coupled vectors $\mathbf{u}_c = [\mathbf{u}_{12}, \mathbf{u}_{13}, ..., \mathbf{u}_{ij}, ..., \mathbf{u}_{m_c}]$
for a given design $\widetilde{\mathbf{d}}$ **do**
  Compute $(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_u, \underline{\mathbf{u}}_c) = \arg\max F(\widetilde{\mathbf{d}}, \mathbf{u}_u, \mathbf{u}_c)$
  for all $\mathbf{u}_{ij} \in \mathbf{u}_c$ **do**
    for all Focal Elements $\theta_{k,ij} \subseteq \Theta_{ij}$ **do**
      $\widehat{F}_{k,ij} = \max_{\mathbf{u}_{ij} \in \theta_{k,ij}} F(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_{\mathbf{u}}, \mathbf{u}_{ij})$
      $\widehat{\mathbf{u}}_{k,ij} = argmax_{\mathbf{u}_{ij} \in \theta_{k,ij}} F$
      Evaluate $bpa(\theta_{k,ij})$
      Evaluate partial Belief curve $Bel(F(\mathbf{u}_{ij}) \leq \nu)$
    end for
    for number of samples **do**
      Evaluate ΔBel^q, $\widehat{\mathbf{u}}_{k,ij}$ and $\widehat{F}_{k,ij}$
    end for
  end for
  for all the combinations of samples **do**
    for all $\mathbf{u}_i \in \mathbf{u}_u$ **do**
      for all Focal Elements $\theta_{k,i} \subseteq \Theta_i$ **do**
        Run
        $F_{max,k,i} = \max_{\theta_{k,i}} F(\widetilde{\mathbf{d}}, \widehat{\mathbf{u}}_{\mathbf{c}}, \mathbf{u}_i)$
        Evaluate $bpa(\theta_{k,i})$
      end for
    end for
    for all the combinations of Focal Elements
    $\theta_t \in \Theta_1 \times \Theta_2 \times ... \times \Theta_{m_u}$ **do**
      Evaluate $F_{max,k} \leq \nu$
      Evaluate bpa_k
    end for
    Evaluate the Belief for this sample by constructing collection Γ_ν
  end for
  Add up all belief values for all samples
end for

The exact curve evaluation requires an optimisation for each FE. With a fraction of optimisation, instead, an approximated Belief curve can be reconstructed with the proposed evidence-network decomposition approach. With reference to Section 7.2.1, it is possible to show the advantage of the method in terms of computational complexity. The complexity is shown in Table 7.1 with respect to the different number of samples and with reference to the number of optimisations required for the exact Belief curve.

As a further advantage, as Fig. 7.1 shows, all the calculated solutions are outer belief approximations since they are a lower bound of the exact one.

Finally, with an increase in the number of samples (proportional to computational

Table 7.1: Complexity comparison of approximated and exact Belief.

n sample	complexity (n opt.)	complexity (%)
1	27	3.7
2	45	6.1
3	63	8.6
4	81	11.1
5	99	13.58
6	117	16.05

Table 7.2: design variables bounds

dim	LB	UB
1	-5	5
2	-5	5
3	-5	5

time) the approximation improves until convergence.

7.2.3 Method Tutorial

The following example serves as a step by step tutorial for the ENM-based decomposition and it is based on the procedure explained in Section 7.2. The test case function is based on topology (b) in Fig. 7.9, reported here in Fig. 7.2 for clarity.

The quantity of interest f is modelled as the sum of three components, f_1, f_2 and f_3, all of them depending on both the design vector $\mathbf{d}$ and the uncertain vector $\mathbf{u}$:

$$f(\mathbf{d}, \mathbf{u}) = f_1(\mathbf{d}, \mathbf{u}) + f_2(\mathbf{d}, \mathbf{u}) + f_3(\mathbf{d}, \mathbf{u}). \tag{7.16}$$

where:

$$f_1 = d_1 + (\sin(u_1) + \sin(u_2)) - (|u_1 + u_2| + \phi_1 + \phi_2)^3 \tag{7.17}$$
$$f_2 = d_2 + (\sin(u_3) + \sin(u_4)) - (|u_3 + u_4| + \phi_1)^3 \tag{7.18}$$
$$f_3 = d_3 + (\sin(u_5) + \sin(u_6)) - (|u_5 + u_6| + \phi_2)^3 \tag{7.19}$$
$$\phi_1 = \sin(u_7) + \sin(u_8) + u_7 + u_8 \tag{7.20}$$
$$\phi_2 = \sin(u_9) + \sin(u_{10}) + u_9 + u_{10} \tag{7.21}$$

The problem consists of 3 design and 10 uncertain variables. The required information is presented in Tables 7.2 and 7.3 respectively. In particular, Table 7.3 list, for each uncertain variable, which of the functions f_i is affected, the domain bounds for each imprecise interval and the corresponding bpa.

In order to apply the decomposition method, it is important first to define the network-based formulation of the model and the partition of the uncertain space into coupling and uncoupling variables. As shown in Fig. 7.2, there is a coupling between

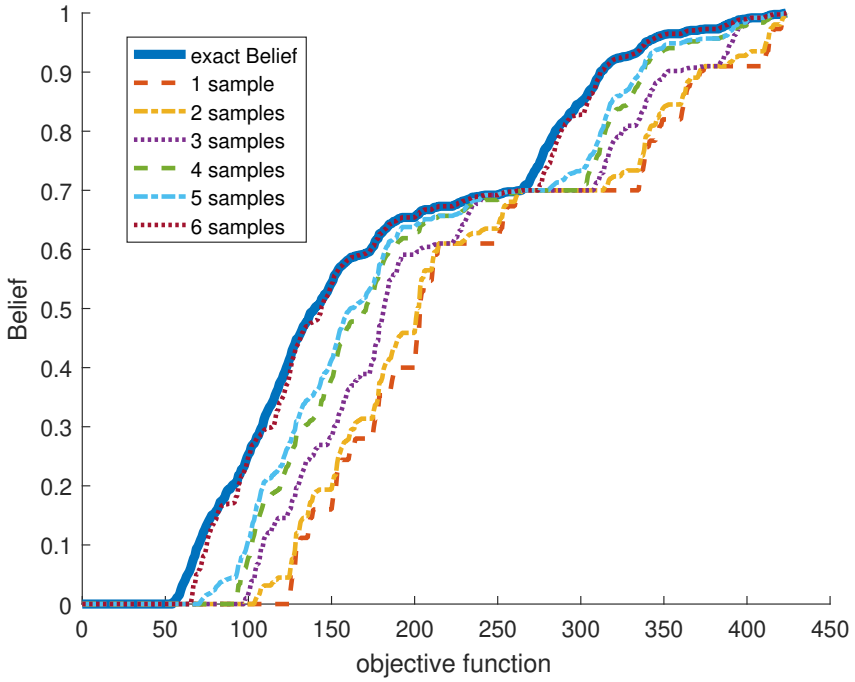


Figure 7.1: Simple example of application of the decomposition approach

Table 7.3: uncertain variables bounds

dim	function	interval 1	bpa 1	interval 2	bpa 2
1	1	[0, 0.5]	0.5	[0.5, 1]	0.5
2	1	[1, 1.5]	0.4	[1.5, 2]	0.6
3	2	[2, 2.5]	0.4	[2.5, 3]	0.6
4	2	[3, 3.5]	0.4	[3.5, 4]	0.6
5	3	[4, 4.5]	0.4	[4.5, 5]	0.6
6	3	[5, 5.5]	0.4	[5.5, 6]	0.6
7	[1, 2]	[6, 6.5]	0.4	[6.5, 7]	0.6
8	[1, 2]	[7, 7.5]	0.4	[7.5, 8]	0.6
9	[1, 3]	[8, 8.5]	0.3	[8.5, 9]	0.2
10	[1, 3]	[9, 9.5]	0.2	[9.5, 10]	0.8

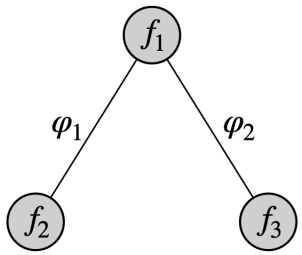


Figure 7.2: topology of the network-model formulation for the tutorial example.

f_1 and f_2 and a further coupling between f_1 and f_3. These couplings are modelled respectively with the exchange functions ϕ_1 and ϕ_2. The uncoupling uncertain space includes the vectors: $\mathbf{u}_1 = [u_1, u_2]$, $\mathbf{u}_2 = [u_3, u_4]$ and $\mathbf{u}_3 = [u_5, u_6]$ affecting respectively the functions f_1, f_2 and f_3. The coupling uncertain space instead includes the vectors: $\mathbf{u}_{1,2} = [u_7, u_8]$ and $\mathbf{u}_{1,3} = [u_9, u_{10}]$ affecting respectively the couples of functions [f_1, f_2] and [f_1, f_3].

The model formulation in Eq. (7.16) can then be extended further by making explicit the dependence of f_1, f_2 and f_3 from uncoupled vectors ($\mathbf{u}_1$, $\mathbf{u}_2$ and $\mathbf{u}_3$) and coupling functions (ϕ_1 and ϕ_2). Also, it is possible to define the dependency of the coupling functions from the coupling uncertain components:

$$f = f_1(\mathbf{d}, \mathbf{u}_1, \phi_1(\mathbf{u}_{1,2}), \phi_2(\mathbf{u}_{1,3})) + f_2(\mathbf{d}, \mathbf{u}_2, \phi_1(\mathbf{u}_{1,2})) + f_3(\mathbf{d}, \mathbf{u}_3, \phi_2(\mathbf{u}_{1,3})) \tag{7.22}$$

Once the problem has been properly formulated, it is possible to start with the decomposition process as described in Section 7.2. The first step for the evaluation of the approximated Belief curve requires the *solution of the optimal worst-case scenario problem* for the definition of the "anchor point". The solution can in general be calculated using the min-max algorithm presented in Chapter 5. For this tutorial, the solution is straightforward: $\mathbf{d}^*$ = [-5, -5] and $\mathbf{u}^*$ = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

The second step requires the *Maximisation over the coupled variables and the computation of* $Bel_c(A)$. This part refers to Eq. (7.4). As shown in Fig. 7.2, there are two coupling functions (ϕ_1 and ϕ_2) depending respectively on the vectors $\mathbf{u_{1,2}}$ and $\mathbf{u_{1,3}}$. Two partial Belief curves need then to be calculated. In particular, $Bel_{c,1}$ (that refers to ϕ_1) requires the maximisation of f over all the FE in Θ_1^c that comes from the Cartesian product of the intervals of variables in $\mathbf{u}_{1,2} = [u_1, u_2]$. Since for both u_1 and u_2 two intervals have been defined in Table 7.3, there are a total of 4 FEs to optimise. All the other uncertain variables - the uncoupled vectors ($\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$) and the coupled vector $\mathbf{u}_{1,3}$ of ϕ_2 - are fixed to the anchor point defined above. Numeric results of the optimisations over the considered FEs for $Bel_{c,1}$ are summarised in Table 7.4. Here, *'id fe'* is the identification number for the FE in Θ_1^c, *'sub-id fe'* indicates for each element of the vector $\mathbf{u}_{1,2}$ which imprecise interval is considered, *'bpa fe'* is the bpa of the FE, *'bpa fe (cum.)'* is its cumulative value considering the FE sorted on *'max f fe'* (their worst-case value of f) and *'max u fe'* is the value of $\mathbf{u}_{1,2}$ corresponding to *'max f fe'*. Likewise, $Bel_{c,2}$ is calculated based only on the 4 FEs in Θ_2^c with all other vector components fixed to the anchor point. Numeric results for the second partial Belief curve

Table 7.4: results of the optimisation of the coupled uncertain components for $Bel_{c,1}$

id fe	sub-id fe	bpa fe	bpa fe (cum.)	max f fe	max u fe
1	[1,1]	0.16	0.16	54	[6.5, 7.5]
2	[2,1]	0.24	0.40	54.5	[7, 7.5]
3	[1,2]	0.24	0.64	54.5	[6.5, 8]
4	[2,2]	0.36	1	55	[7, 8]

Table 7.5: results of the optimisation of the coupled uncertain components for $Bel_{c,2}$

id fe	sub-id fe	bpa fe	bpa fe (cum.)	max f fe	max u fe
1	[1, 1]	0.06	0.06	54	[8.5, 9.5]
2	[2, 1]	0.14	0.20	54.5	[9, 9.5]
3	[1, 2]	0.24	0.44	55	[8.5, 10]
4	[2, 2]	0.56	1	55	[9, 10]

are summarised in Table 7.5. The calculated $Bel_{c,1}$ and $Bel_{c,2}$ curves are then plotted in Fig. 7.3. In the figure, the blue lines correspond to Belief and Plausibility of the contribution of ϕ_1 to f while the red ones refer to the contribution of ϕ_2 to f. It has to be noticed that the worst condition of both $Bel_{c,1}$ and $Bel_{c,2}$ in Fig. 7.3 correspond to the solution of the min-max optimisation.

The third step is the *Sampling of the partial curves* $Bel_{c,i}$. This is described in Eq. (7.6). In this example, 2 samples have been used for both the curves $Bel_{c,1}$ and $Bel_{c,2}$ in Fig. 7.3. The approach selects the points in Fig. 7.3 starting from the worst-case solution (Belief = 1) and then moving down with a reduction of the threshold in the belief. Given the sample in Fig. 7.3, it is then possible to determined the corresponding FE as in Tables 7.4 and 7.5, and in particular, the values for the worst case f and the corresponding $\mathbf{u}_c$ that makes f maximum inside the FE.

The fourth step implies the *maximisation over the uncoupled variables* as described Eq. (7.7). A belief (resp. plausibility) curve is computed for each function f_i and each combination of samples. From the previous step, 2 samples are given for each $Bel_{c,i}$, for a total of 4 combinations. Fig. 7.4 shows 4 plots, each one obtained for a different sample. Each plot shows belief and plausibility of the 3 functions f_i. In this test-case two of these curves happens to be coincident. As can be noted from the pictures, the cumulative values of the bpa are scaled. The scaling coefficient for the q-th sample is the ΔBel^q explained in Eq. (7.9) calculated as the product of the differences in Fig. 7.3 between the belief value of the sample and the previous sample.

Finally, the last step requires the *reconstruction of the approximation* $\widetilde{Bel}(A)$. This is explained in Eqs. (7.8) and (7.9). The reconstructed outer approximation for both belief and plausibility is in Fig. 7.5. In Fig. 7.6 also shows the exact belief and plausibility calculated with a optimisation (maximisation for the Belief and minimisation for the Plausibility) for each FE.

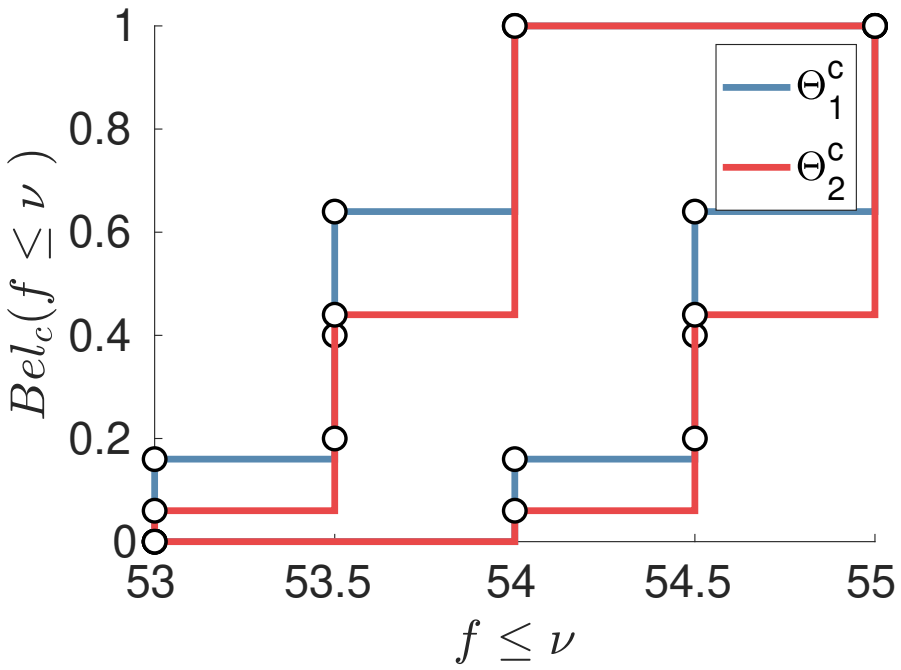


Figure 7.3: Simple example of application of the decomposition approach

7.2.4 Benchmark and test results

Finally, this section presents the benchmark that has been developed for the testing of *Belief* and *Plausibility* outer approximation. The set of test cases is shown in Fig. 7.9 and Eq. (7.23). The results are listed in Table 7.6.

The benchmark is defined based on the analytical function f: $f = \sum_i g_i$.

The i-th node' performance and the generic coupling function h_{ij} between couples of nodes are respectively formulated as:

$$g_i = d_i + \sum_{k=1}^{N^u_{FE,i}} \sin u_{ik} - \left(\left| \sum_{k=1}^{N^u_{FE,i}} u_{ik} \right| + \sum_{k=1}^{m_c} h_{ik} \right)^3 \tag{7.23}$$

$$h_{ik} = \sum_{k=1}^{N^u_{FE,i}} \sin\left(u_{ik}\right) + u_{ik}$$

f is multi-modal with respect of Θ_u and Θ_c separately while monotonic with respect of the coupling functions h.

The scaling behaviour of the method is tested by applying f to the different network topologies represented in Fig. 7.9. In particular, each i-node depends on a pair of uncoupling uncertain variables $\mathbf{u}_{u,i} = [u^1_{u,i}, u^2_{u,i}]$ and each link shares between two nodes a pair of coupling uncertain variables $\mathbf{u}_{c,ij} = [u^1_{c,ij}, u^2_{c,ij}]$. Then, being n the number of nodes and l the number of links for a selected topology, the total number of

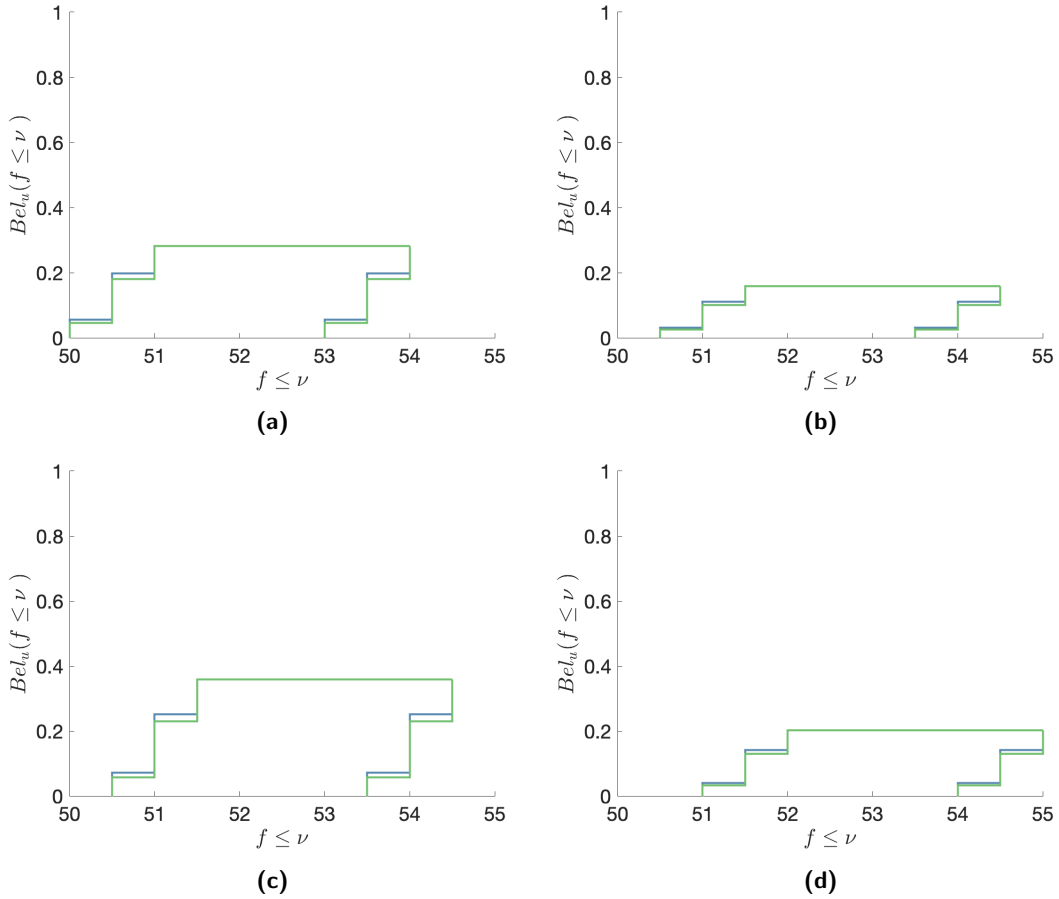


Figure 7.4: Belief curves due to the uncoupled uncertain vectors for each of the 4 combinations of samples.

uncertain variables is: $n_{u,tot} = 2(n + l)$.

First, the optimal design configuration $\mathbf{d}_{opt}$ is calculated by solving the constrained min-max optimisation problem described by Eq. (5.1) by means of Algorithm 2. Fig. 7.8 presents, as an example, the convergence of the algorithm for f applied to topology (c) where the constraint function has been defined as:

$$C(\mathbf{d}, \mathbf{u}) = -\sum_i d_i + 5 \sum_i \big(\sin(d_i u_i)\big) \leq 0 \tag{7.24}$$

The figure shows the convergence of the algorithm to the optimal solution trading the conflict between the performance indicator

$$f_{max} = \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \tag{7.25}$$

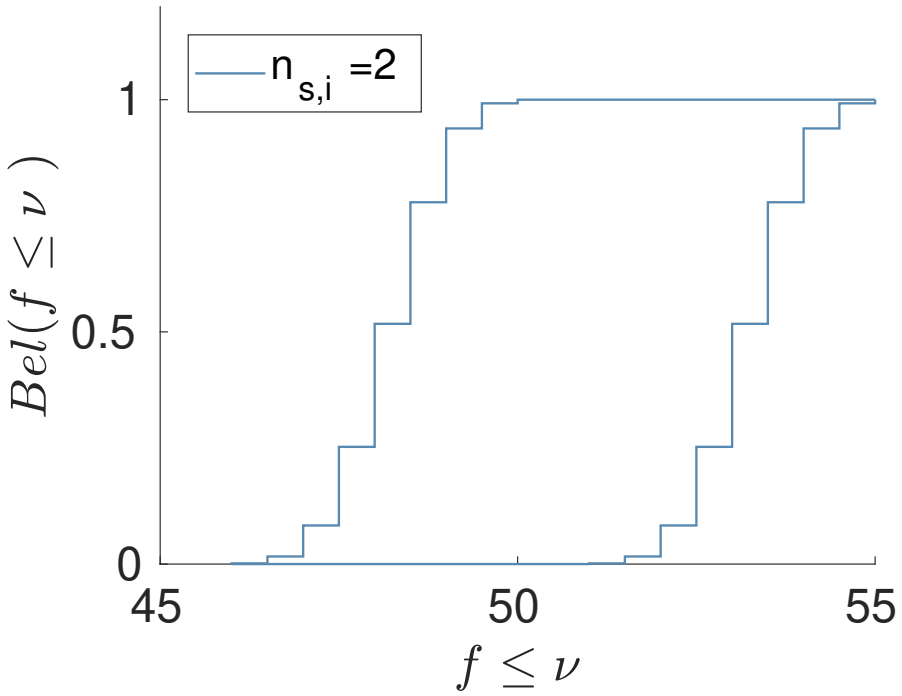


Figure 7.5: Simple example of application of the decomposition approach

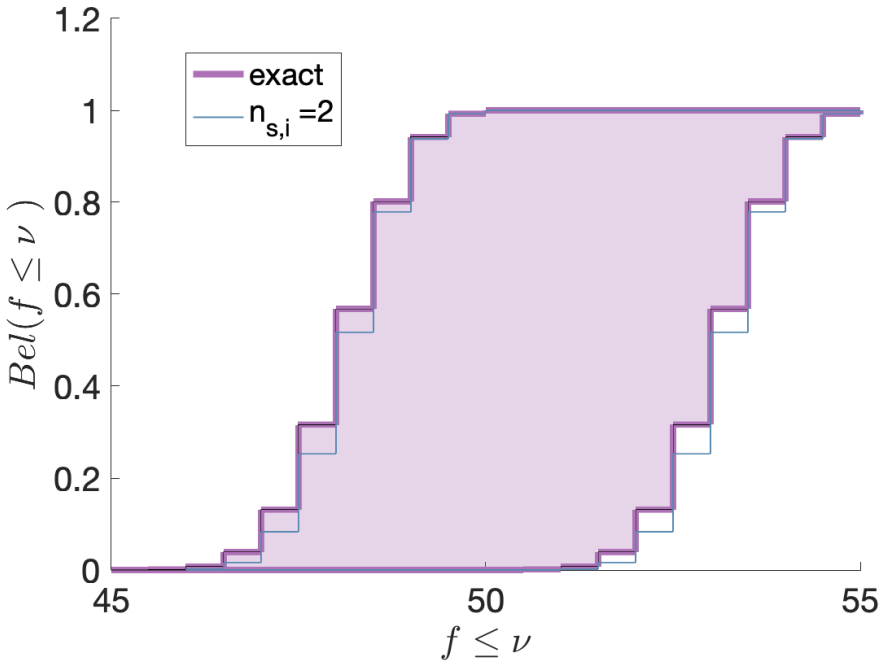


Figure 7.6: Simple example of application of the decomposition approach

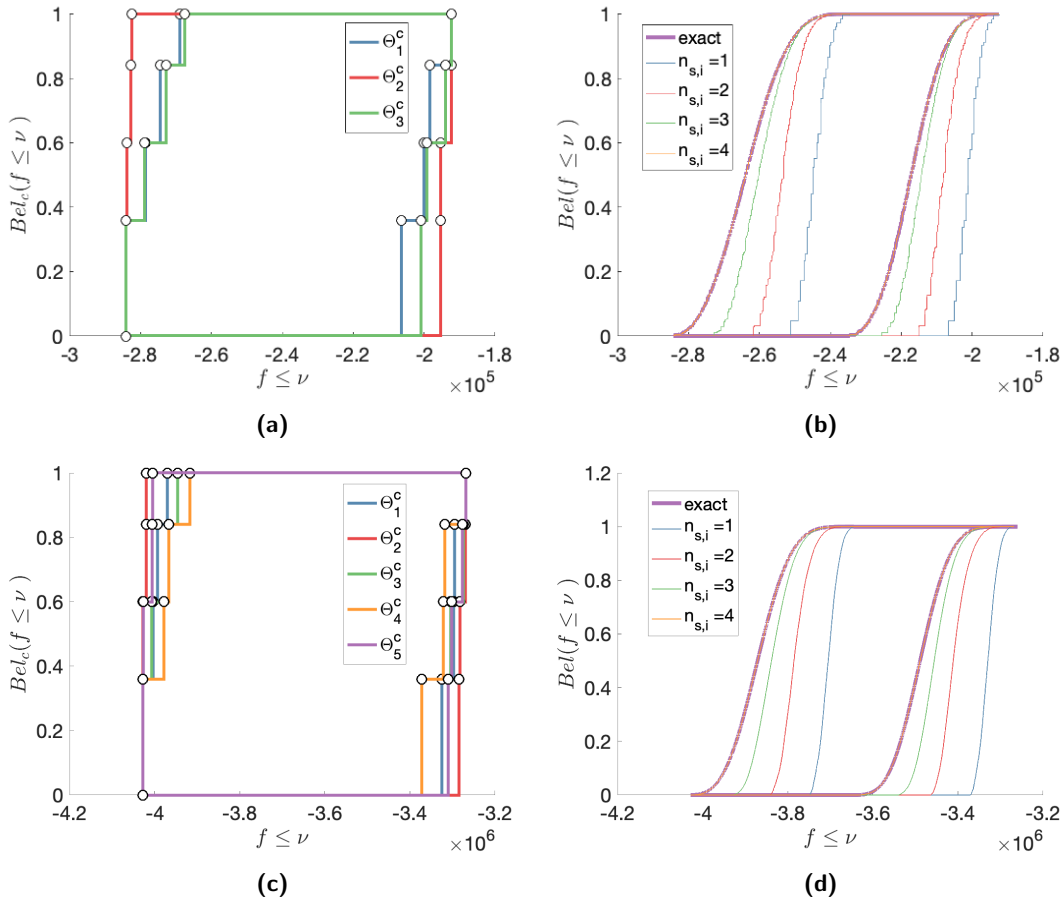


Figure 7.7: Belief and Plausibility curves for f. Sub-figures (a,b) refer to topology (c) while sub-figures (c,d) refer to topology (d). Sub-figures (a,c) plot the partial curves evaluated only in the subspace of the coupling uncertain variables. Each colour corresponds to a single link in the network. Sub-figures (b,d) show the final curves calculated with the decomposition approach where each colour refers to a different sampling. They show also the exact belief and plausibility evaluated running an optimisation for each focal element.

and the constraint function

$$c_{max} = \max_{\mathbf{u} \in U} c(\mathbf{d}, \mathbf{u}) \leq 0 \tag{7.26}$$

At convergence, $\mathbf{d}_{opt}$ gives the minimum worst case value of f_{max} while pushing c_{max} at the edge of the feasible set $]-\infty, 0]$. The curve c, instead, represents the value of the constraint function corresponding to $\mathbf{d}_{opt}$ and the worst uncertain scenario for f.

The network topology is then exploited with Algorithm 11 to propagate epistemic uncertainty and calculate at a reduced computational cost the Belief and Plausibility curves corresponding to $\mathbf{d}_{opt}$. For this purpose, the Frame of Discernment is constructed by assigning to each uncertain variable u two possible intervals and their bpa. The total

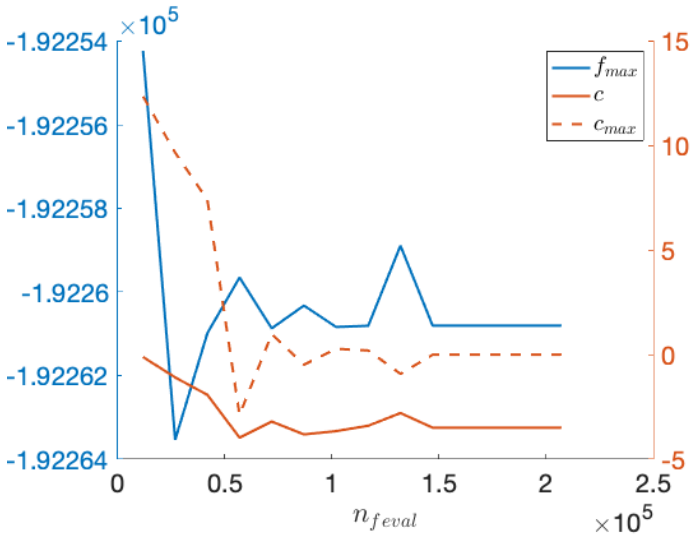


Figure 7.8: Convergence to the optimal solution of the constrained min-max problem for Algorithm 2 with f applied to topology (c). For each design solution proposed by the algorithm at each new iteration, it is here plotted the worst-case scenario in the uncertain space for the objective function f_{max} and for the constraint violation c_{max}. It is also plotted the value of the constraint c corresponding to the worst scenario for f_{max}.

number of focal elements is then $n_{fe} = 2^{n_{u,tot}}$. The number of optimisations required for exact quantification of the Belief curve is $n_{opt,exact} = n_{fe}$ while the number required by the decomposition approach follows Eq. (7.12).

Table 7.6 collects the results of the simulations for the different network topologies. It shows the gain in computational cost offered by the decomposition approach together with the generated error. The error has been evaluated as the ratio $(A_{exact} - A_{dec})/A_{dec}$ where A_{exact} and A_{dec} are the integral of the exact and decomposition curve respectively. For problems satisfying the conditions in Section 6.1, the decomposition approach assures quantifying exactly the DST' measures of probability when the partial curves are entirely sampled. For example, considering the topology (c) and using 64 samples (all the combinations for the 4 samples for each of the 3 coupled curves), we obtain an error equal to zero running a number of optimisations that is 19% of a total number of focal elements. Furthermore, the smaller the number of samples, the lower the computational cost but the higher the error. For example for the same problem, a single sample brings to a cost that is 0.59% of the exact evaluation increasing the error however to 187%. Between these two extreme positions, we can make a trade-off between cost and accuracy.

Fig. 7.7 shows instead the plots of the partial curves and the final curves calculated with the decomposition strategy and also the exact curves calculated running an optimisation for each focal element for f applied to topology (c).

It has been noticed that sampling very close points in the space of the maxima in Fig. 7.7(a) brings to negligible contribution to the error reduction in Fig. 7.7(b). Then, future works will show how a careful selection of the samples can improve the algorithm

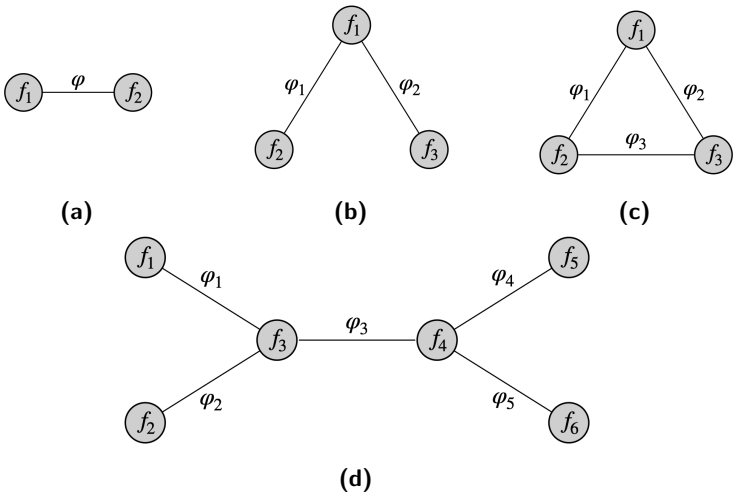


Figure 7.9: Network topology applied to TC_1 and TC_2 for the study of the scalability of the decomposition method. (a): simple graph with a dimension of the uncertain space $dim_u = 6$. (b): triad $\wedge$ with $dim_u = 10$. (c): triangle $\mathbb{K}_3$ with $dim_u = 12$. (d): graph with $dim_u = 22$.

performance mitigating both the error (possibly bringing it to zero) and the cost under the condition that the user is not interested in the entire curve but only in the value assumed by belief and plausibility at some specific thresholds. This is particularly important when the decomposition strategy is nested within an optimisation loop as in Eq. (7.1).

7.3 Case Study: Optimal Battery Sizing

The EBRO is here tested on a realistic case study of space systems engineering. This case study will address the battery sizing in order to assure autonomy of a spacecraft in the transfer to a Geostationary Orbit (GEO).

The mass of the battery is dependent on the following design, uncertain and fixed parameters:

- **3 design parameters**. Time of orbit insertion t, type of battery γ and bus voltage ΔV_{bus}:

$$\mathbf{d} = \left[t, \gamma, \Delta V_{bus}\right]^T.$$

  Table 7.7 shows the design parameters with their associated range of variability: t is given in Modified Julian Date (MJD) - for more details see [194] - and can be any day of 2019 at 7 a.m.; γ ranges in four possible intervals - [0, 0.25) , [0.25, 0.5), [0.5 0.75) and [0.75, 1] - corresponding to 4 battery types (see Table 7.8).

- **31 uncertain parameters**. Orbital parameters of five orbits $\boldsymbol{\alpha}$ - semimajor axis, **a**, eccentricity, **e**, Inclination, **i**, right ascension of the ascending node, $\boldsymbol{\Omega}$, Argument of Perigee, ω and True Anomaly, θ (for more details see [194])- and

Table 7.6: ENM decomposition results

topology	n_{fe}	$n_{samp,i}$	$n_{samp,tot}$	n_{opt}	n_{opt}/n_{fe}	cpu f (s)	cpu fmincon (s)	cpu Alg. (s)	Belief error
(a)	$2^6 = 64$	1	1	12	0.1875	0.008	0.341	0.4265	0.46635
		2	2	20	0.3125	0.01	0.312	0.4685	0.40311
		3	3	28	0.4375	0.018	0.377	0.5570	0.20439
		4	4	36	0.5625	0.019	0.448	0.6415	2e-7
(b)	$2^{10} = 1024$	1	1	20	0.0195	0.014	0.387	0.662	1.524
		2	4	56	0.0546	0.031	0.692	0.930	4.344
		3	9	116	0.1133	0.053	1.137	1.435	0.1618
		4	16	200	0.1953	0.089	1.896	2.288	0.0001
(c)	$2^{12} = 4096$	1	1	24	0.0059	0.017	0.428	0.732	1.8705
		2	8	108	0.0264	0.0525	1.061	1.353	0.59611
		3	27	336	0.0820	0.151	2.95	3.492	0.14550
		4	64	780	0.1904	0.346	6.62	7.650	0.00004
(d)	$2^{22} = 4194304$	1	1	44	1.049e-05	0.0495	0.694	1.17	2.6101
		2	32	788	1.878e-04	0.611	7.202	10.717	0.5776
		3	243	5852	1.395e-03	4.32	48.50	73.38	0.1870
		4	1024	24596	5.864e-03	17.88	200.28	299.21	0.0002

Table 7.7: design parameters

variable	symbol	lower bound	upper bound	var. type
insertion time	t	6939.8	7304.8	continuous
battery type	γ	0	1	discrete
Bus voltage	ΔV_{BUS}	0	5	discrete

Table 7.8: lookup table of batteries

BATTERY	A	B	C	D
Cell capacity C_{cell} (Ah)	4.5	1.7	1.5	3.7
Cell voltage ΔV_{cell} (V)	4.1	4.2	4.2	4.1
Cell Mass $m(\gamma)$ (kg)	0.63	0.2	0.21	0.23
Max DoD DoD_{max} (%)	80	75	75	75

efficiency of the battery η:

$$\mathbf{u} = [\boldsymbol{\alpha}, \eta] = [\mathbf{a}, \mathbf{e}, \mathbf{i}, \boldsymbol{\Omega}, \boldsymbol{\omega}, \boldsymbol{\theta}, \eta]$$

with:
$\mathbf{a} = [a_1, a_2, a_3, a_4, a_5]^T$,
$\mathbf{e} = [e_1, e_2, e_3, e_4, e_5]^T$,
$\mathbf{i} = [i_1, i_2, i_3, i_4, i_5]^T$,
$\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \Omega_3, \Omega_4, \Omega_5]^T$,
$\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]^T$,
$\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^T$.

For each uncertain variable, two possible intervals are given, both with 50% of probability. They are symmetrically arranged on either side of the nominal values in Table 7.10; the interval dimensions are given by Table 7.9.

Table 7.9: uncertain intervals

	a (km)	e (-)	i (°)	Ω(°)	ω (°)	θ (°)
Δu	±20 km	±0.0012	±0.07°	±30°	±0.5°	±0.025°

- **10 fixed parameters**. Engine ignition time and Liquid Apogee Engine (LAE) burn time per orbit (Table 7.11).

The input is the time of insertion by the launcher in MJD, plus a table with as many rows as LAE firings plus one (Table 7.10) - the first row corresponds to insertion by the launcher. Columns specify the orbital parameters after the burn, the time of arrival, in hours from launch and the duration of burn. The spacecraft is assumed to follow Keplerian motion between burns. Note that the consistency of the input is not checked, i.e. it is assumed that the specified burn can deliver to the specified ephemerides within the specified time

The overall mass M depends on the mass of the cell (Table 7.8) and on the total number of cells of the battery:

$$M = m(\gamma) N_{tot} \tag{7.27}$$

and N_{tot} is the product of the number of cells in series N_s and in parallel N_p:

$$N_{tot} = N_s N_p = \frac{\Delta V_{BUS}}{\Delta V_{cell}} \frac{E}{DoD_{max} \Delta V_{BUS} C_{cell}} + 1 \tag{7.28}$$

where E, the Energy requirement, depends on the length of the eclipse periods evaluated from the input data, ΔV_{BUS} is the Bus voltage, ΔV_{cell} the cell voltage, DoD_{max} the maximum allowed Depth of Discharge, C_{cell} the Cell Capacity [195]. It is considered that the battery is ON during eclipse and LAE burns and OFF otherwise.

7.3.1 Problem Formulation

The optimisation problem is formulated as the constrained minimisation of the mass M of the battery under epistemic uncertainty.

Table 7.10: Nominal value of the epistemic parameters for SSTL problem

orbit	a (km)	e (-)	i (°)	Ω (°)	ω (°)	θ (°)
1	68500.3	0.90	22.81	86.63	180.10	0.00
2	73250.2	0.77	9.12	86.79	180.6	180.08
3	86065.5	0.51	1.09	85.96	180.81	180.84
4	49646.4	0.15	0.36	86.85	180.97	4.25
5	42049.0	0.001	0.05	270	0.00	359.95

Table 7.11: Fixed parameters

orbit	Time of Arrival [hrs.]	Burn duration [hrs.]
1	0.00	0.0
2	24.7	0.6
3	79.9	0.8
4	114.7	0.6
5	145.2	0.3

First the constraint min-max problem is solved were M is the total mass:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} M(\mathbf{d},\mathbf{u}) \\ &s.t. \\ &\forall \mathbf{u} \in U: \quad c(\mathbf{d},\mathbf{u}) \leq 0. \end{aligned} \tag{7.29}$$

where the constraint is given by the maximum allowed Depth of discharge for each type of battery (Table 7.8):

$$c(\mathbf{d},\mathbf{u}) = DoD(\mathbf{d},\mathbf{u}) - DoD_{max} \tag{7.30}$$

Then the $Belief$ and $Plausibility$ curves are reconstructed to quantify the effect of uncertainty on the solutions.

7.3.2 Evidence Network Model of the Battery Sizing Problem

The above-described battery sizing problem is modelled with an ENM assuming that each of the five orbits is a node of the network and all five nodes are connected to a 6^{th} node, the battery. The scheme of Fig. 7.10 illustrates this simple topology:

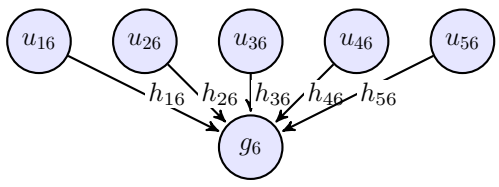


Figure 7.10: Evidence Network Model of the optimal battery sizing problem.

$$M(\mathbf{d},\mathbf{u}) = \sum_{i=1}^{6} g_i(\mathbf{d},\mathbf{u}_i,\mathbf{h}_i(\mathbf{d},\mathbf{u}_i,\mathbf{u}_{ij})) \tag{7.31}$$

- 5 nodes evaluate the energy required by the spacecraft in the given orbits: $E_i(\mathbf{d}, a_i, e_i, i_i, \Omega_i, \omega_i, \theta_i)$ and have $g_i = 0$ for $i = 1, 2, ..., 5$
- The 6th node sizes the mass of the battery: $g_6(\mathbf{d}, \eta, h_{16}, h_{26}, h_{36}, h_{46}, h_{56})$ where $h_{i6} = E_i$.

The epistemic vector has been organised as:

$$\mathbf{u} = [\underbrace{\mathbf{u}_6}_{\text{uncoupled}}, \underbrace{\mathbf{u}_{16}, \mathbf{u}_{26}, \mathbf{u}_{36}, \mathbf{u}_{46}, \mathbf{u}_{56}}_{\text{coupled}}]$$

where: $\mathbf{u}_6 = \eta$ and $\mathbf{u}_{i6} = [a_i, e_i, i_i, \Omega_i, \omega_i, \theta_i]^T$ with i = 1,...,5.

$\Theta_6 = \cup_k \theta_{k,6} \ni \mathbf{u}_6$ is the set of all the FE of the uncoupled vector $\mathbf{u}_6$ and $\Theta_{i6} = \cup_k \theta_{k,i6} \ni \mathbf{u}_{i6}$ is the set of all the FE of the coupled vector $\mathbf{u}_{i6}$ for i=1, 2, ..., 5.

In the ENM model of this problem, the five orbits independently contribute to the mass as the uncertainty on the energy requirement manifests only through node 6. Furthermore, node 6 is monotone with respect to the energy requirement of each orbit, independently of the uncertainty in the other orbits. Finally, the mass of the battery M is proportional to the maximum energy requirement that depends on the maximum period of battery discharge. Because of the monotonic dependency of the discharge period on the uncertainty in each orbit, the maximum energy demand can be calculated directly from the min-max solution. Thus:

$$M \propto E^{max} \Rightarrow M \propto \max(E_1^{max}, E_2^{max}, E_3^{max}, E_4^{max}, E_5^{max}). \tag{7.32}$$

With the specific orbital parameters used in this case study, the min-max algorithm always converges to a solution where the maximum energy requirement derives from the second orbit; thus only the second node of the presented ENM influences the sixth, the battery, through the exchange function h_{26}: $g_6(\mathbf{d}, \eta, h_{26})$ where $h_{26} = E_2$.

7.3.3 Results

The software MP-AIDEA [131] has been used to provide the solution of both the min-max and the h-decomposition problems. The inner loop (maximisation over $\mathbf{u}$), the outer loop (minimisation over $\mathbf{d}$) and also each optimisation for the decomposition approach have been set with a single population and a maximum number of function evaluations $N_F^{max} = 1000$ while the total number of function evaluations for whole the min-max loop is $N_F^{max,tot} = 10^5$; the problems have been run multiple times obtaining the same results.

Min-Max

The minmax solution for the optimal battery sizing is $\widetilde{d} = [59, D, 36.9]^T$ with a corresponding battery mass of 126.3 kg. For $t = 59$ the mission is most affected by

uncertainty, as shown in Figs. 7.11a and 7.11b. The Figures show the influence of the uncertainty for all the possible missions in the year 2019. The blue curves correspond to the nominal orbits in Table 7.10, while the red ones represent the maximisation over the uncertain parameters of the energy requirement (Fig. 7.11a) and time of eclipse (Fig. 7.11b).

From the analysis results that the total required energy is not affected by the uncertain parameters for a number of dates. In fact, for a mission that starts on a day $\in [1, 33] \cup [90, 225] \cup [276, 365]$, nominal energy is equal to the maximum energy (1540 Wh) and they depend only on the LAE burn duration. However, from day 34 to day 89 and from day 226 to 275 the energy requirement is strongly influenced by the uncertainties on the orbits and day 59 is certainly the most affected. From the minmax solution battery D results to be the best one for all the possible mission times. Fig. 7.12 shows that the mass (nominal and maximum) of battery D is the lowest for all days of the year.

H-Decomposition

The full Belief curve of the battery sizing problem requires $2^{31} = 2.1475 \cdot 10^9$ maximisations (one for each FE), thus it is intractable. The computational cost of the decomposition approach, as explained in Eq. (7.13), is:

$$N_{FE}^{DEC} = N_s \sum_{i=1}^{m_u} N_{FE,i}^{u} + \sum_{i=1}^{m_c} N_{FE,i}^{c} = 320 + 2N_s. \tag{7.33}$$

Fig. 7.13a shows the partial Belief curve of the mass M considering all the FE $\theta_{k,26} \in \Theta_{26}$ of the coupled variables $\mathbf{u}_{26}$ that influence both sub-systems two and six; the other partial curves are not significant for the final reconstruction because, as explained in Section 7.3.2, the uncertainty on the nodes one, three, four and five (and the corresponding orbits) have no influence on the value of the objective function M. Finally Fig. 7.13b shows the reconstructed Belief curve obtained with the h-decomposition approach with five samples in the space of the coupled vectors $\mathbf{u}_{i6}$. Thus, according to Eq. (7.33), 330 different optimisation have been run, strongly reducing the computational cost of the exact evaluation.

7.3.4 Validation

A validation of the correctness of the results can be obtained in this way. Decompose the min-max problem in three steps:

- Fix a starting day for the mission: $\hat{t}$;
- Maximise the energy requirement over the orbit uncertainty:

$$\max_{\mathbf{u} \in U} E(\hat{t}, \mathbf{a}, \mathbf{e}, \mathbf{i}, \boldsymbol{\Omega}, \boldsymbol{\omega}, \boldsymbol{\theta})$$

- Minimise the mass over the design parameters (type of battery and voltage):

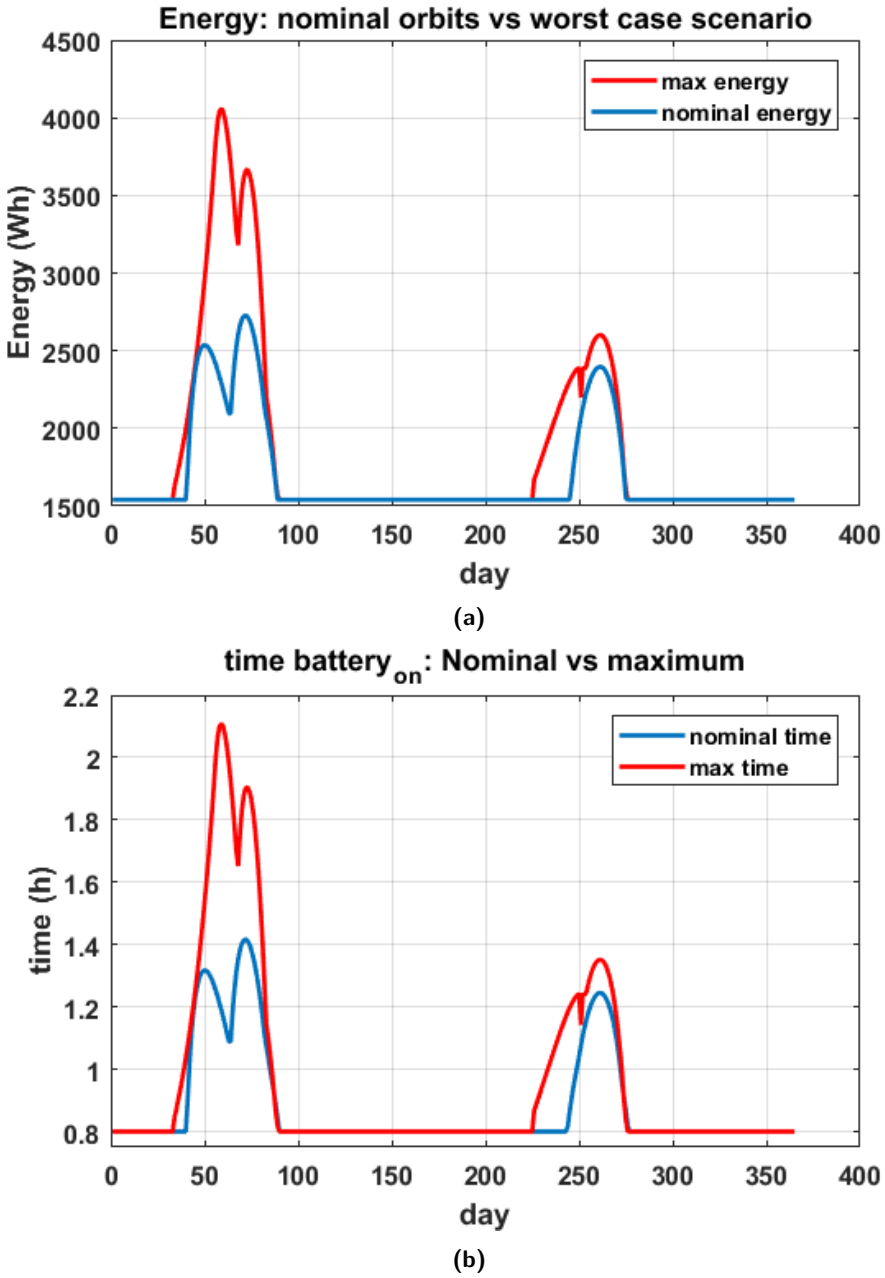


Figure 7.11: First analysis: each day of 2019 has been considered for the satellite launch; for each day, the nominal and worst case scenario have been evaluated for the energy requirement(Figure 7.11a) and time of eclipse (Figure 7.11b).

$$\min_{\substack{\mathbf{d}\in D \\ t=\hat{t}}} M(\mathbf{d}, \mathbf{a}, \mathbf{e}, \mathbf{i}, \boldsymbol{\Omega}, \boldsymbol{\omega}, \boldsymbol{\theta}, \widehat{\eta})$$

with $\widehat{\eta} = \min(\eta)$.

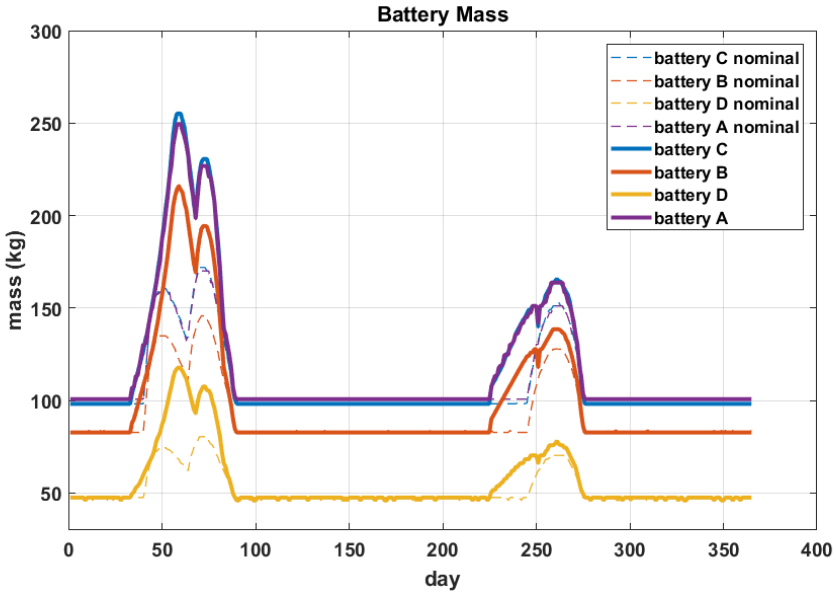
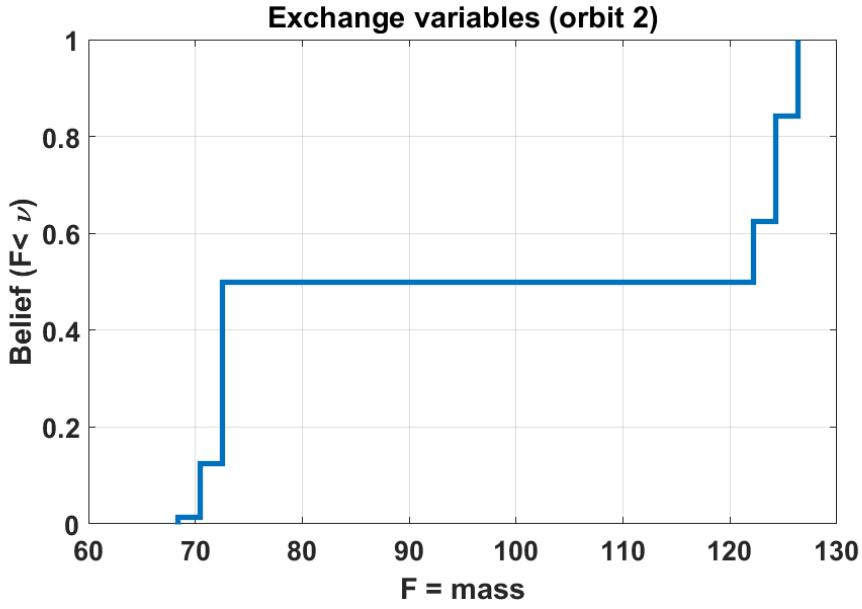


Figure 7.12: Comparison of the nominal and maximum masses obtained with the four different batteries for each day of 2019.

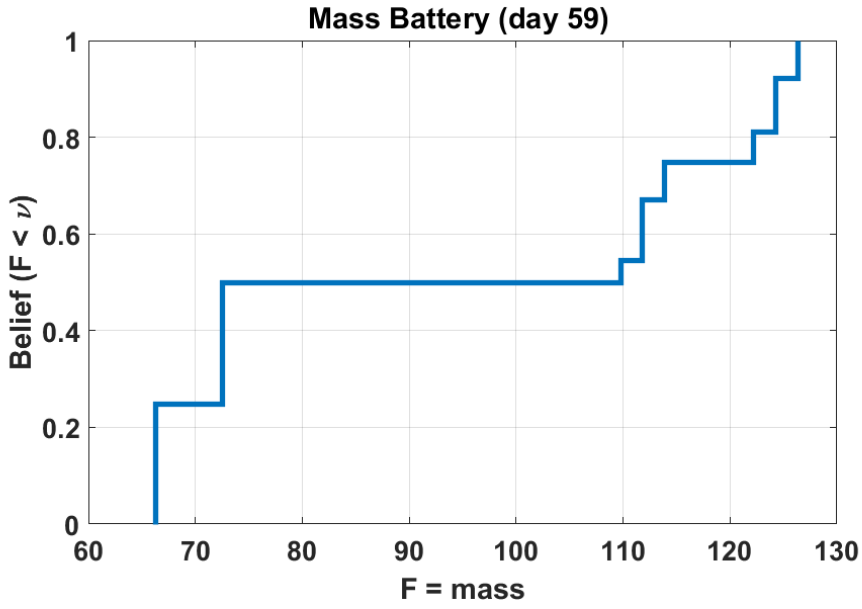
For $\hat{t} = 59$ and evaluated $\arg\max_{\mathbf{u}\in U} E$ (the red curve in Fig. 7.11a, the minimum Mass of the battery (Fig. 7.12) corresponds to battery D and it is 126.3 kg.

7.3.5 Comments

In this case study the energy requirement and the corresponding mass of a battery have been analysed in order to assure autonomy of a spacecraft in the transfer to a geostationary orbit GEO. The mission is composed of five different orbits and four LAE firings. Our model, with the given information, is able to evaluate the periods of time when the satellite is in eclipse and then the mass of the battery in order to assure enough storage of energy. Firstly, the energy requirement has been studied for each day of the year 2019 on the nominal given data and on the worst case over the uncertainty on the orbits of the satellite. It was noticed that there are two periods in the year, 28.5% of the entire period, where the uncertainty plays an important role in the determination of the mass of the battery. For the remaining 71.5% of the time, instead, given the previously described information and evidence, the uncertainty does not influence the quantity of interest and the mass of the battery can be evaluated deterministically. In order to make an interesting analysis, we decided to choose, for the launch, two days when the uncertainty on the position of the spacecraft during the orbits could be dangerous for the mission: day 59 and day 261. For both days we evaluated the worst-case scenario, that is the best configuration over the design parameters d that minimises the negative effects of the uncertainty u. Finally, fixed the design variables (type of battery and bus voltage) from the min-max solution, we reconstructed the Belief (lower probability) that the battery, with a mass $M < \eta$, is able to satisfy the



(a)



(b)

Figure 7.13: Partial Belief curve, Figure 7.13a, of the coupled vector $\mathbf{u}_{26}$ and Belief curves, Figure 7.13b, of the spacecraft' mass for the day 59 obtained with 5 samples and then 330 optimisations.

energy requirement. Our analysis shows that, given the information and the evidence described in the previous sections, battery D has the minimum mass for each day of the year (Figure 7.12). Due to supplier confidence issues rather than technical performance the actual SSTL design is battery A. Figure 7.14 shows two comparisons. The first is

between the belief curve of battery A and the belief curve of battery D in the day 59 (resp. yellow and purple curves): for that day, when the influence of uncertainty is important, the mass of battery A has to be much heavier than battery D in order to be successful in the worst case scenario. The second comparison regards battery A: red, blue and purple curves represent respectively: the belief curve in the 72% of the days (of the year 2019), the actual design choice and the belief curve of battery A in day 59. The red Belief is simply a line because in these configurations (days 1-33, 90-225 and 276-365) the uncertainty doesnt influence the mass of the battery. Red and purple curves are the lower and upper bounds for the belief of battery A during the year. In the 72% of the days the belief curve for battery A is the red plot, in the 5% of the day the belief curve of battery A is to the right of the red plot but still to the left of the blue threshold while only in the 23% of the days the belief curve is to the right of the actual design (blue); in the worst day, 59, it is the purple one. This means that the actual design (blue) has a Belief=1 to be successful for each mission starting on the 77% of the days of the year 2019.

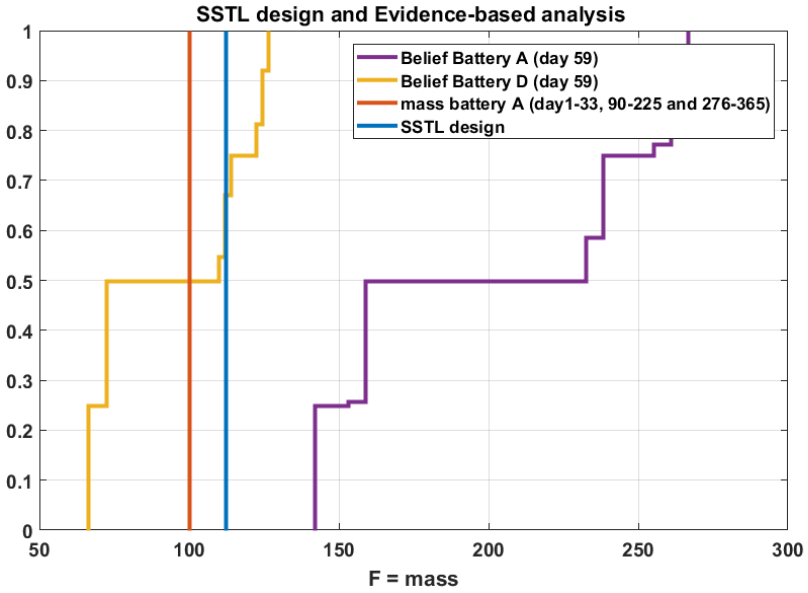


Figure 7.14: Comparison between the SSTL design (battery A, blue curve), the worst case scenario for the battery A in the 72% of the time of the year (red curve) and the belief curves for both batteries A and D in day 59 (purple and yellow curves).

7.4 Conclusion

The chapter presented an application of EBRO to the optimisation of a complex space system affected by epistemic uncertainty. We used DST of Evidence to model uncertainty and ENM to model the system and its interconnections. From the battery sizing problem one can see that: the proposed constrained min-max scheme produces correct

results and the decomposition approach delivers an approximation of the Belief nearly in polynomial time with a low exponent. Once the computational cost of individual subsystems become important compared to the overall evaluation of system performance and reliability, an approach based on hierarchical surrogate models can be used, as demonstrated in [116].

The chapter presented also a novel tool for UQ in the framework of DST which is useful when IP is used to model lack of knowledge and imprecision. The proposed approach is iterative, produces a conservative approximation (converging from below) and drastically reduces the computational complexity. The method is based on the system decomposition where the whole system model is decomposed in coupled submodules with exchanging functions. It is based on the ENM which translates a complex engineering system into a non directed graph. In this graph each node represents a sub-system or component and the associated uncertainty is quantified with basic belief assignments. Exploiting the ENM properties, a decomposition procedure is explained that reduces the computational cost to a polynomial complexity with respect to the number of interacting components. An extensive description of the procedure has been presented. A benchmark of increasing complexity has also been proposed with which the approach has been tested.

8

Evidence-Based Resilience Optimisation

" It is not the most intellectual of the species that survives;
it is not the strongest that survives;
but the species that survives is the one that is able best to adapt and adjust to the changing environment in which it finds itself. "
– Charles Darwin, On the Origin of Species, 1859

The content of this chapter was published in:

- **G. Filippi**, M. Vasile, D. Krpelik, P. Z. Korondi, M. Marchi, and C. Poloni, Space systems resilience optimisation under epistemic uncertainty, *Acta Astronautica*, vol. 165, pp. 195210, 12 2019, https://doi.org/10.1016/j.actaastro.2019.08.024 [13].
- **G. Filippi** and M. Vasile, Network resilience optimisation of complex systems in *Advances in uncertainty quantification and optimization under uncertainty with aerospace applications (M. Vasile and D. Quagliarella, eds.)*, (Cham), Springer International Publishing, 2020, *https : //doi.org/*10.1007/978 − 3 − 030 − 80542 − 5 [57].
- **G. Filippi**, D. Krpelik, P. Z. Korondi, M. Vasile, M. Marchi, and C. Poloni, Space systems resilience engineering and global system reliability optimisation under imprecision and epistemic uncertainty, in *Proceedings of the International Astronautical Congress, IAC, vol. 2018-Octob, (Bremen)*, 2018 [64].
- **G. Filippi** and M. Vasile, Evidence-based resilience engineering of dynamic space systems, in *Proceedings of the International Astronautical Congress, IAC*, vol. 2019-Octob, (Washington), 2019 [65].
- **G. Filippi** and M. Vasile, Network Resilience Optimisation of Complex Systems in *International Conference on Uncertainty Quantification & Optimisation*, pp. 111, 2020 [59].

- **G. Filippi**, D. Gillespie, A. Ross Wilson, and M. Vasile, A resilience approach to the design of future Moon base power systems, in *Int. Astronaut. Congr. IAC*, 2020 [69].

As the complexity of a system grows - being it a natural, engineering or organisational system - the associated risk of bad performances, failures and even disasters will increase as well. After an hazardous events, the (complex) system survives if it is able to absorb disturbances and shocks and then to adapt itself to the new environment. In the attempt to solve these problems within the engineering field, Design for Resilience in the context of systems engineering is acquiring continuously more attention in these last years. Resilience Engineering is a relatively recent field of research that has blended together different disciplines [24–27]. Like all the new research directions, its terminology and definitions are still in their infancy [23,28]. Having experts from many different fields with highly variegate backgrounds also don't help in finding easily an agreement. This slow progression is however important due to the very nature of resilience which is strictly connected to the concepts of complexity and holism. Resilience is indeed an emergent property of complex systems which exhibits similar aspects in much different applications. It requires then the development of strong multidisciplinary theoretical foundations to be used successfully in practice.

The chapter proposes a novel algorithmic methodology for the resilience optimisation of a CEdS called EBORe. Our proposed concept extends and integrates the concepts of Design for Reliability and Design for Robustness and introduces the use of DST to model epistemic uncertainty. The idea is that a resilient system should be able to endure disturbances and recover from shocks [24, 26, 27, 81] while maintaining an optimal level of performance and functionalities. In other words, the system is expected to transition between different potentially degraded states but without losing the ability to maintain or recover, in full or in part, its functionalities and performance. In this framework, the aim of resilience engineering is to maximise performance and resilience at the same time. This can be translated into finding the design solution that maximises the level of performance and active functionalities under the effect of uncertainty that affects the transition to multiple states. The ability to endure disturbances can be engineered by maximising robustness. In particular, one could be interested in the worst case scenario in which the effect of uncertainties is maximum. In mathematical terms, robustness can be translated into a deterministic min-max optimisation problem [11] that aims at optimising performance in the worst case scenario. This aspect is here complemented with the ability to recover after shocks. A shock can be seen as a probabilistic transition to a degraded state. A system reliability model is then introduced to quantify these transitions and relate them to the design solutions.

Two mathematical modelling framework are then presented to quantify the system resilience. The first is based on stochastic processes and it is in particular a HCTMC approach that discretise a finite set of possible system's states. The second is based on bifurcation theory and it is able to continuously model the dynamics of the system's functionality.

The applicability of the proposed method to space systems engineering is demonstrated through the preliminary design of a small satellite in Low Earth Orbit (LEO).

The goal of the satellite is to take pictures of the Earth and send them to a ground-based station. The system is supposed to be affected by epistemic uncertainty.

Summarising, the contribution of this chapter is: i) a definition of Resilience Engineering, ii) a mathematical framework for EBORe, iii) two mathematical models to quantify resilience and iv) an algorithmic procedure for resilience engineering that combine methodologies previously presented in earlier chapters.

The reader can find the necessary background for this chapter in Chapters 2, 5 and 6. Chapter 2 introduces to engineering approaches for design optimisation and to resilience engineering in particular. The chapter introduces also to stochastic processes with focus on Markov Chain and to the Bifurcation Theory. Chapter 5 presents the algorithmic methodology for the solution of the constraint min-max problem. Chapter 6 presents the modelling framework based on the ENM.

The remaining of this section presents the structure of the chapter. Section 8.1 defines our approach to resilience engineering. Section 8.2 defines the mathematical formulation of the problem. Sections 8.3 and 8.4 then present two alternative mathematical models of system resilience: the former is based on a HCTMC while the latter uses Bifurcation Theory. Section 8.5 is an application to a realistic test-case with the fist resilience model. Section 8.6 presents instead an application for the second resilience model. Results are discussed. Section 8.8 finally concludes the chapter.

8.1 Resilience Engineering

Resilience is here defined as the ability of a system to endure disturbances or regain a desirable operational state after the occurrence of a shock. The former characteristic of resilience is directly connected to the robustness of the system. Hence in the following, we will propose an approach to enhance robustness when the possible disturbances are captured by a model of epistemic uncertainty. The latter characteristic of resilience can be quantified by measuring the degree of recovery of a system indicator, over time, after a failure [24]. We will propose a global system reliability model that relates the epistemic uncertainty in system and environmental parameters and the design choices to the transition between different functioning states. Thus, our concept of Resilience Engineering, combines robustness and reliability with a time component that accounts for the temporal variation of system performance and the response to disturbances and shocks.

The uncertainties in system characteristics and environment are deemed to be epistemic in nature and are modelled with DST as the underlying assumption is that they cannot be captured by a known probability distribution. This uncertainty model is applied to a graph representation of the space system, i.e. the ENM. We then quantify the values of the performance indexes of the ENM by propagating the effects of the epistemic uncertainties through the network and the global reliability model.

We finally apply an optimisation framework to identify, under uncertainty, those design choices that maximise performance, and the time dependent reliability when disturbances and possible multiple disruption and recovery events occur. The maximisation of system's resilience implies the minimisation of the risk of occurrence of

catastrophes while leaving the possibility, during the mission time, to have partial failures.

8.2 Problem Formulation

Two types of metrics are considered in EBORe. One is the performance function f:

$$\begin{aligned} f\colon \mathcal{R}^{n+m+1} &\to \mathcal{R} \\ [\mathbf{d},\mathbf{u},t]^T &\mapsto f(\mathbf{d},\mathbf{u},t) \end{aligned} \tag{8.1}$$

The other is the system functionality g:

$$\begin{aligned} g\colon \mathcal{R}^{n+m+1} &\to \mathcal{R} \\ [\mathbf{d},\mathbf{u},t]^T &\mapsto g(\mathbf{d},\mathbf{u},t). \end{aligned} \tag{8.2}$$

Both the performance f and the functionality g depend on the time $t \in \mathbb{T} \subset \mathcal{R}$, a design vector $\mathbf{d} \in D \subset \mathcal{R}^n$ and a uncertain vector $\mathbf{u} \in U \subset \mathcal{R}^m$.

We propose to formulate the resilient design problem of a generic CEdS as a constrained worst-case optimisation where f is the objective function and g the constraint:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}) \\ &s.t. \\ &\nu - \min_{\mathbf{u}\in U} g(\mathbf{d},\mathbf{u}) \leq 0. \end{aligned} \tag{8.3}$$

In Eq. (8.3) uncertainty is treated with a deterministic worst-case approach. Following the modelling framework in Chapter 6, the mathematical models f and g of the CEdS can be formulated as a ENM. The *robustness* is guaranteed by the min-max optimisation of the performance f under epistemic uncertainty. The problem is then constrained with the satisfaction, in the worst scenario, of the functionality g which incorporate the global reliability model. The combination of *robustness* and *reliability*, finally, guarantees the *resilience* of the solution.

8.3 Markov Chain-Based Resilience Model

In this section we introduce a stochastic method for modelling possible functionality impairments and restorations based on a HCTMC approach. The method can be used to study the resilience of complex systems and will be here applied, in particular, to space systems. We assume a random occurrence of both *disasters* and *repairs* during the satellite mission. The satellite is modelled as a (finite) multi-state system and its performance, both instantaneous and cumulative, depends on its state and trajectory.

We denote the set of possible states of the satellite by $\mathcal{X}$ and the satellite trajectory in this state space by a stochastic process $X : \mathbb{T} \to \mathcal{X}$, where $\mathbb{T}$ is the temporal dimension. A stochastic process is uniquely determined by an initial distribution over the state space, say P_0, and a family of conditional distributions, the transition operators, $\{P(X(t)|X(s))\}$ where $\{s,t\} \in \mathbb{T}$.

In the case of HCTMC processes, the specification can be simplified [196]. HCTMC is uniquely determined by its *transition rate matrix*, $Q \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$, which is an analogue to the derivative in the theory of ordinary differential equations. If the non-diagonal elements of a transition rate matrix are non-negative and the sum of elements in each row is zero, it induces a family of transition operators of the form:

$$P(X(t) = x|X(s) = y) = \exp(Q(t-s))(y, x), \tag{8.4}$$

where exp denotes a matrix exponential. The probability of obtaining state x at time t, can then be evaluated by:

$$P(X(t) = x) = \sum_{y\in\mathcal{X}} P_0(y) \exp(Qt)(y, x). \tag{8.5}$$

Suppose that our functionality measure, which is to be optimised, is a cumulative functionality, $V_T = \int_0^T V(t)dt$, over the mission time T, and that the immediate functionality $V(t)$ depends on the state of the satellite at the respective time, $X(t)$. Since X is a stochastic process, $V(t)$ and the cumulative functionality V_T become random variables. In order to formulate a real valued function for the optimisation problem, we need to take the stochastic character of V_T into account. The function can be replaced by a real functional on the underlying probability space. We choose it to be the expected value, thus the functionality g becomes:

$$g(\mathbf{d}, \mathbf{u}) := \mathbb{E}\left\{\int_0^T V(t, X(t); \mathbf{d}, \mathbf{u})dt\right\}, \tag{8.6}$$

where $V(t, X(t); \mathbf{d}, \mathbf{u})$ emphasises the dependency of the immediate performance on the system state, a set of design parameters (or design choices) $\mathbf{d}$, and a set of uncertain parameters $\mathbf{u}$. Due to the Fubini's theorem [197], we can switch the order of integration to obtain:

$$g(\mathbf{d}, \mathbf{u}) = \int_0^T \mathbb{E}\left\{V(t, X(t); \mathbf{d}, \mathbf{u})\right\} dt. \tag{8.7}$$

Because the set of system states, $\mathcal{X}$, is finite, Eq. (8.7) attains its final form:

$$g(\mathbf{d}, \mathbf{u}) = \int_0^T \sum_{x\in\mathcal{X}} \left\{V(t, x; \mathbf{d}, \mathbf{u})P(X(t) = x)\right\} dt. \tag{8.8}$$

Eq. (8.8) implies that we can calculate the function g in two steps. First, solve the stochastic process $X(t)$, and second, integrate the performance with pre-calculated values of $P(X(t))$. If the immediate performance function is defined to be discrete in time, the integration into the expected cumulative performance in Eq. (8.8) will become a summation with respect to a counting measure.

8.4 Catastrophe Theory-Based Resilience Model

As presented in Chapter 6, we suggest to model the CEdS using the ENM approach where the system is translated into a graph, the p sub-systems to its nodes and the system's structure to the network's topology.

Then, our goal is to study the coupling between topology and system dynamics. We suggest modelling the dynamics of each component's functionality with Bifurcation Theory [198, 199]:

$$\dot{x} = h_{\mu}(x). \tag{8.9}$$

Indeed, Eq. (8.9) allows us to capture the continuous transition between fully functioning and degraded states and the occurrence of disruptions and shocks that perturb the system. The bifurcation parameter μ in Eq. (8.9) is responsible for qualitative changes in the dynamics.

Such a model can be developed for each node in the network (sub-system of the CEdS). Finally, all these models can be coupled to understand the global dynamics of the system. One possible coupling approach [200] directly creates a functional dependency between states of different nodes. For the generic component i,

$$\dot{x}_i = W_{\mu_i}\left(x_i(t)\right) + \sum_{j=1}^{p} A_{ij} Q_{\nu_{ij}}\left(x_i(t), x_j(t)\right) \tag{8.10}$$

provides a rather general deterministic description of systems governed by pairwise interactions. The first term on the right-hand side of Eq. (8.10), $W_{\mu_i}(x_i)$, describes the self-dynamics of x_i, accounting for processes such as influx, degradation or reproduction. The second term captures the interactions of node i with its neighbours. In particular, A_{ij} is the adjacency matrix and $Q_{\nu_{ij}}(x_i, x_j)$ describes the dynamical mechanism governing the pairwise interactions. In Eq. (8.10), both W_{μ_i} and $Q_{\nu_{ij}}$ are bifurcation models.

A different approach could instead define the bifurcation parameters μ_i or the initial condition $x_{0,i}$ of the Ordinary Differential Equations (ODEs) as function of the output of a different node j:

$$\mu_i = R(f_j(\mathbf{d}, \mathbf{u})) \tag{8.11}$$

or

$$x_{0,i} = R(f_j(\mathbf{d}, \mathbf{u})) \tag{8.12}$$

with f_j the output of system j.

Then, by using one of this coupling mechanisms, the whole system of ODEs from in Eq. (8.10) can be summarised as:

$$\dot{\mathbf{x}} = h_{\boldsymbol{\mu}}(\mathbf{x}). \tag{8.13}$$

with $\mathbf{x}, \boldsymbol{\mu} \in \mathcal{R}^p$. Both $\mathbf{x}_0$ and $\boldsymbol{\mu}$ in Eq. (8.13) depend in general on the design vector

$\mathbf{d} \in \mathcal{R}^n$, the uncertain vector $\mathbf{u} \in \mathcal{R}^m$ and time t. For the simple case $p = q = 1$, it is:

$$\begin{aligned} x, x_0, \mu \colon \mathcal{R}^{n+m+1} &\to \mathcal{R} \\ [\mathbf{d}, \mathbf{u}, t]^T &\mapsto x(\mathbf{d}, \mathbf{u}, t) \\ [\mathbf{d}, \mathbf{u}, t]^T &\mapsto x_0(\mathbf{d}, \mathbf{u}, t) \\ [\mathbf{d}, \mathbf{u}, t]^T &\mapsto \mu(\mathbf{d}, \mathbf{u}, t), \end{aligned} \tag{8.14}$$

The reliability function ρ is then evaluated by normalising the solution $\mathbf{x}$ in Eq. (8.13) as it is explained in Section 8.4.2:

$$\begin{aligned} \rho \colon \mathcal{R}^{n+m+1} &\to [0, 1]^T \\ [\mathbf{d}, \mathbf{u}, t]^T &\mapsto \rho(\mathbf{d}, \mathbf{u}, t), \end{aligned} \tag{8.15}$$

where $\rho = 1$ indicates a system fully functioning and $\rho = 0$ a system with a non recoverable failure. More precisely, the functionality g combines the selected QoI with the dynamics of ρ:

$$g(\mathbf{d}, \mathbf{u}, t) = \int_{T_0}^{T_M} \text{QoI}(\mathbf{d}, \mathbf{u}, t)\rho(\mathbf{d}, \mathbf{u}, t)dt. \tag{8.16}$$

In the following, the bifurcation approach for the resilience modelling will be applied only to one node (sub-system). This implies that we are solving a simplified version of Eq. (8.13) that is Eq. (8.9). Future work will extend the approach to the solution of a system of ODEs as suggested in Eqs. (8.10) and (8.12).

8.4.1 Autonomous Bifurcation

We present here some basic one-dimensional bifurcation models. Plots are in Fig. 8.1. The behaviour of real dynamical systems can be captured by a combination or variation of these elementary building blocks.

- Tangential bifurcation. A tangential bifurcation happens when one stable and one unstable equilibria points collide and annihilate when varying the bifurcation parameter μ:

  $$\dot{x} = \mu - x^2 \tag{8.17}$$

  Eq. (8.17) presents the critical point at $\mu_c = 0$. For $\mu > 0$ there are two equilibria: $x = \pm\sqrt{\mu}$ with the positive one stable and the negative one unstable. For $\mu < 0$ instead, there are no stable equilibria. $x^* = 0$ is finally a non-hyperbolic equilibrium. Looking at the bifurcation diagram in Fig. 8.1a we can however say that it is a saddle point and then it is unstable.

- Trans-critical bifurcation. With a trans-critical bifurcation, the equilibria of the system exchange stability as the parameter μ crosses the critical value $\mu_c = 0$. Consider the equation:

  $$\dot{x} = \mu x - x^2 \tag{8.18}$$

  It has two equilibria points: $x^* = 0$ and $x^* = \mu$. The former is stable if $\mu < 0$

and unstable if $\mu > 0$. The latter is stable if $\mu > 0$ and unstable if $\mu < 0$. $\mu = 0$ is a critical saddle unstable point.

As for the tangential bifurcation, also for the trans-critical one it holds $f_{\mu_c}(x^*) = 0$ with $f'_{\mu_c}(x^*) = 0$. However it is also: $\frac{\partial f_\mu}{\partial \mu}(x_c) = 0$. The bifurcation plot is in Fig. 8.1b.

- Pitchfork bifurcation. There are two types of pitchfork bifurcations. In the super-critical one,

$$\dot{x} = \mu x - x^3. \tag{8.19}$$

a stable equilibrium, passing through the critical point $\mu_c = 0$, becomes unstable generating other two stable equilibrium points as shown in Fig. (8.1c). Instead, in the sub-critical Pitchfork bifurcation,

$$\dot{x} = \mu x + x^3 \tag{8.20}$$

when $\mu < 0$ the dynamical system presents one stable and two unstable equilibria that, passing through the critical point $\mu_c = 0$, collapse generating an unstable equilibrium as shown in Fig. 8.1d.

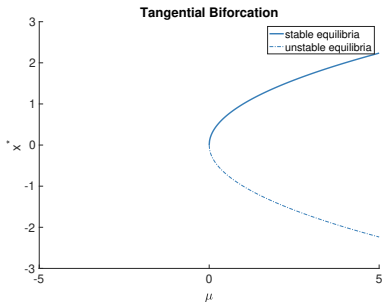
- Bifurcation with Hysteresis. An interesting phenomenon in Bifurcation theory is the hysteresis. It happens when, for a fixed parameter μ there exist more than one attractors. Consider, for example, the family of differential equations

$$\dot{x} = \mu + x - \frac{1}{3}x^3. \tag{8.21}$$

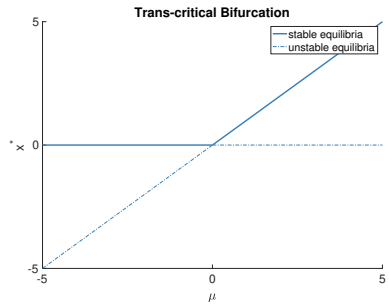
It presents two stable equilibrium sets for $-5 \leq \mu \leq \frac{2}{3}$ and for $\frac{2}{3} \leq \mu \leq 5$ and one unstable equilibria set for $-\frac{2}{3} \leq \mu \leq \frac{2}{3}$. As Fig. (8.1e) shows, there is an overlapping between the three sets. In particular, for a fixed μ $s.t.$ $-\frac{2}{3} \leq \mu \leq \frac{2}{3}$, the system converges to different stable solution depending on the initial state x_0.

- Hopf bifurcation: A Hopf Bifurcation occurs when a periodic solution or limit cycle, surrounding an equilibrium point, arises or goes away as a parameter μ varies. When a stable limit cycle surrounds an unstable equilibrium point we have a *super-critical* Hopf bifurcation. When, instead, an unstable limit cycle surrounds a stable equilibrium point we have a *sub-critical* Hopf bifurcation. With a change of coordinates from Cartesian to polar, the super-critical and the sub-critical Hopf bifurcations can be represented by Fig. 8.1c, 8.1d respectively. Fig. 8.1f is instead a 3D phase plot of the super-critical Hopf bifurcation given by the Lienard equation:

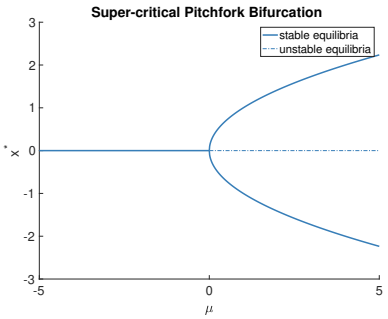
$$\ddot{x} - (\mu - x^2)\dot{x} + x = 0. \tag{8.22}$$



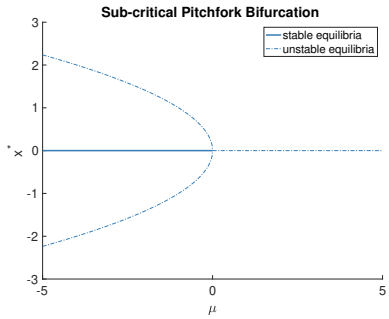
(a) Tangential Bifurcation in Eq. (8.17)



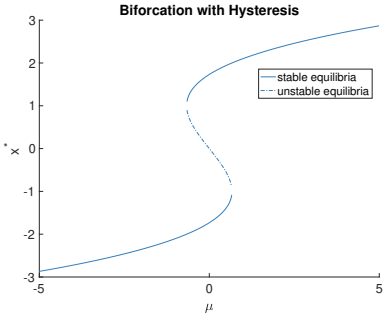
(b) Trans-critical Bifurcation in Eq. (8.18)



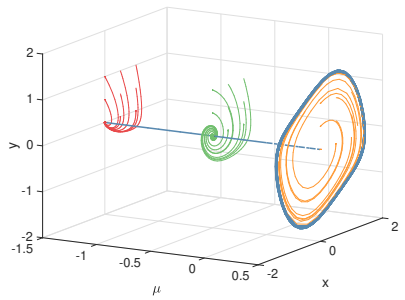
(c) Super-critical Pitchfork bifurcation in Eq. (8.19)



(d) Sub-critical Pitchfork bifurcation in Eq. (8.20)



(e) Bifurcation with Hysteresis in Eq. (8.21)



(f) super-critical Hopf Bifurcation in Section 8.4.1

Figure 8.1: Sample of local autonomous bifurcations diagrams.

Table 8.1: normalisation parameters

parameter	description	examples
x_{inf}	fully functional state	-5.5
x_{sup}	total failure state	5.5
$K_{-\infty}$	form factor	-1e2
K_{∞}	form factor	1e2
ϵ_ρ	minimum survival state	1e-2

8.4.2 Normalisation

This section describes the normalisation procedure used to map $x(t)$ in Eq. (8.13) to $\rho(t)$ in Eq. (8.15). We first have to define which value x_{inf} corresponds to the total failure $\rho = 0$ and which value x_{sup} corresponds to a total functional state $\rho = 1$:

$$x_{inf} = \min_{\mathbf{d}\in D,\mathbf{u}\in U} x_0(\mathbf{d},\mathbf{u}) + K_{inf} \tag{8.23}$$

$$x_{sup} = \max_{\mathbf{d}\in D,\mathbf{u}\in U} x_0(\mathbf{d},\mathbf{u}) + K_{sup}. \tag{8.24}$$

with K_{inf} and K_{sup} form factors that allow us to adapt the reliability function $\rho(t)$ shifting and stretching it. We define also for which values ($K_{-\infty}$ and K_{∞}) the solution is considered to diverge. In particular:

$$x(t)\begin{cases}\text{is divergent} & \text{if } x(t) < K_{-\infty} \wedge x(t) > K_{\infty} \\ \text{is not divergent} & \text{if } K_{-\infty} \leq x(t) \leq K_{\infty}\end{cases} \tag{8.25}$$

The following equation is finally used to evaluate the function ρ:

$$\rho(t) = \min\left[\frac{\max\left[x(t)k(t) - x_{inf}, 0\right]}{x_{sup} - x_{inf}}, 1\right] b(t) \tag{8.26}$$

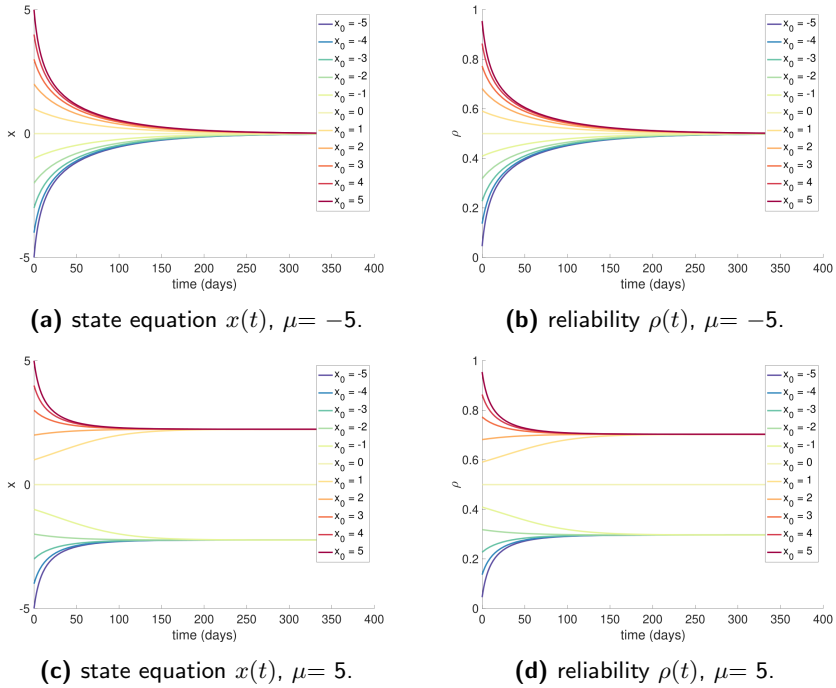
where $k(t)$ is a correction factor that modifies the solution when the state x diverges to $+\infty$:

$$k(t) = \begin{cases}1 & \text{if } x(t) < K_{\infty} \\ \text{-1} & \text{if } x(t) \geq K_{\infty}\end{cases} \tag{8.27}$$

and $b(t)$ take into account the minimum accepted level for ρ:

$$b(t) = \begin{cases}1 & \text{if } \rho(t) > \epsilon_\rho \\ 0 & \text{if } 0 \leq x(t) \leq \epsilon_\rho\end{cases} \tag{8.28}$$

Summarising, the approach has the following characteristics: (i) it normalises the dynamics x mapping it to ρ in the interval $[0, 1]$; (ii) in both cases when x diverges to $+\inf$ or $-\inf$, the reliability ρ goes to zero; (iii) the threshold ϵ_p has a similar effect to the non-recovery state in HCTMC: when $\rho < \epsilon_p$ it is not possible to recover the system state and we consider this as the total failure.



(a) state equation $x(t)$, $\mu = -5$. **(b)** reliability $\rho(t)$, $\mu = -5$.

(c) state equation $x(t)$, $\mu = 5$. **(d)** reliability $\rho(t)$, $\mu = 5$.

Figure 8.2: Evolution in time of the state x (a,c) and the normalised state ρ (b,d) with different initial points x_0. The plots correspond to the super-critical pitchfork bifurcation in Eq. (8.19) with different values of the parameter μ.

Table 8.1 lists the parameters and values used for the normalisation in Figs. 8.2 to 8.7. The results have been calculated for $x_0 \in [-5,\ 5]^T$ and for the two extreme values of the parameter $\mu = [-5,\ 5]^T$. From the figures, we see that the scalarisation preserves the qualitative behaviour of the curve. The reliability function assumes values $0 \leq \rho \leq 1$. When the solution x diverges for both positive or negative values, ρ becomes zero. If the system reduces ρ below the minimum accepted ϵ_ρ, then there is no possibility to recover.

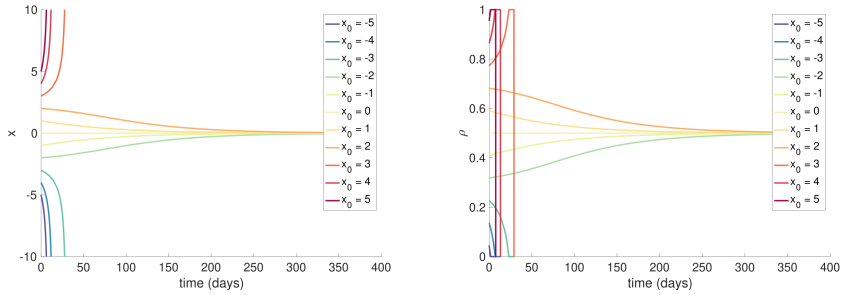
8.4.3 Non Autonomous Bifurcation: $\mu(t)$

The autonomous bifurcation models presented in Section 8.4.1 can be generalised. As an example, this section shows four non-autonomous models for which the bifurcation parameter $\mu(t)$ is time-dependent. The reader can find more details in [65].

1. Smooth Degradation and Recovery Model. The reliability ρ is modelled with the super-critical pitchfork bifurcation in Eq. (8.19), where the parameter μ is:

$$\mu(t) = 13.8\mu_0 t \sin(10t)\text{sign}(x_0) \tag{8.29}$$

with $\mu_0 = \mu(t = 0)$ the initial value of the parameter μ. As μ assumes a new value, the stable and unstable equilibria change according to Fig. 8.1c. The model is plotted in Fig. 8.4 with a set of different initial conditions.



(a) state equation $x(t)$, $\mu = -5$ **(b)** reliability $\rho(t)$, $\mu = -5$

(c) state equation $x(t)$, $\mu = 5$ **(d)** reliability $\rho(t)$, $\mu = 5$

Figure 8.3: Evolution in time of the state x (a,c) and the normalised state ρ (b,d) with different initial points x_0. The plots correspond to the sub-critical pitchfork bifurcation in Eq. (8.20) with different values of the parameter μ.

2. Shock Model. This model is also a generalisation of the super-critical pitchfork bifurcation in Eq. (8.19), for which a discontinuity in ρ for both loss or recovery is introduced. The effect of a shock on the system can be simulated with an abrupt variation of the value of μ over time:

$$\mu(t) = \begin{cases} \mu_0 & \text{if } t < \max(10, 30|x_0|) \\ -x_0^2\mu_0 & \text{if } t \geq \max(10, 30|x_0|). \end{cases} \tag{8.30}$$

The model is plotted in Fig. 8.5. We can see that the parameter $\mu(t)$ allows us to a switch between the different equilibrium points of Fig. 8.1c. This switch can be slow or fast depending on the magnitude of the discontinuity introduced by $\mu(t)$. The variation of the state can go in both the directions modelling both degradation and recovery.

3. Shock and Recovery Model. We model here a shock followed by a recovery where the recovery is still possible `iff` the shock does not bring to a total failure ($\rho > \epsilon_\rho$ after the shock). Considering again the Eq. (8.19), the parameter μ is here modelled as

$$\mu(t) = \begin{cases} \mu_0 & \text{if } t < 20 \vee t > 20 + |x_0 - 5| \\ -2x_0|5 + x_0||\mu_0| & \text{if } 20 \leq t \leq 20 + |x_0 - 5| \end{cases} \tag{8.31}$$

4. Shock and Repair Model. The use of Eqs. (8.29) to (8.31) to model $\mu(t)$, brings to a limitation. The dynamics of the equations define some closed areas from which the states x and ρ can not escape: as in Fig. 8.4b, a system with an initial state $0.5 \leq \rho \leq 1$ will always remain in that interval while a total failure can happen only if the initial state is $0 \leq \rho < 0.5$. This limit can be overcome with the use of a bifurcation with hysteresis. Considering Eq. (8.21), we can model μ as

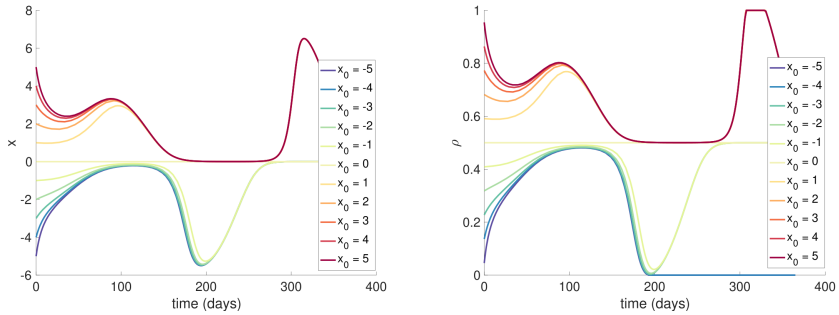
$$\mu(t) = \begin{cases} \mu_0 & \text{if} \\ -2x_0|5 + x_0||\mu_0| & \text{if} \end{cases} \tag{8.32}$$

The results are plotted in Fig. 8.7.

A further generalisation of these non-autonomous bifurcation models, used in the following test cases, is given by the dependency of the bifurcation parameter μ from the design $\mathbf{d}$ and the uncertain $\mathbf{u}$ vectors of variables: $\mu(\mathbf{d}, \mathbf{u}, t)$.

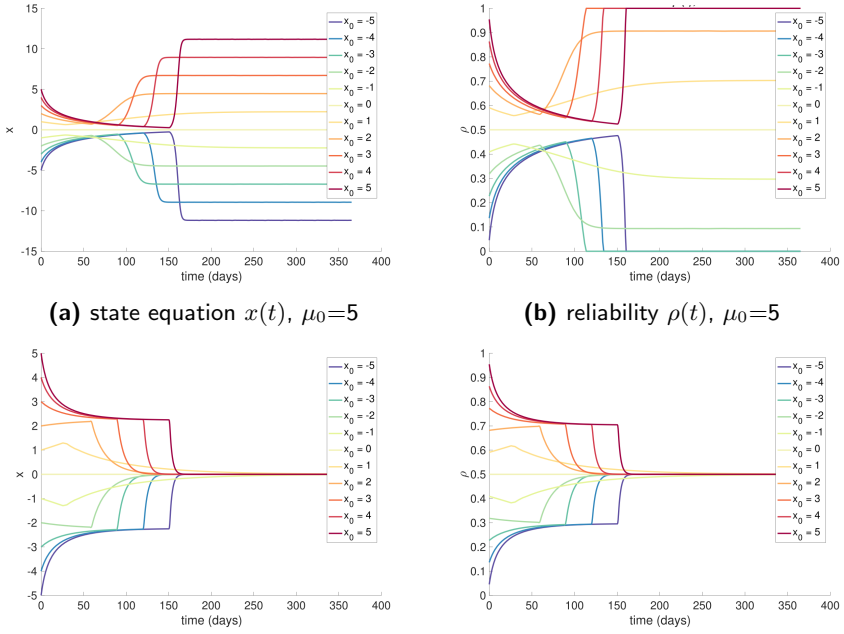
8.5 Test Case Application (Markov Model)

This section presents the application of EBORe to the preliminary design of a small satellite in LEO where the resilience model is based on HCTMC as explained in Section 8.3. The goal of the satellite is to take pictures of the Earth and transmit them to a ground based receiver. The satellite is assumed to be composed of 5 subsystems: AOCS, TTC, OBDH, Power and Payload. Each component has multiple functionalities and both the performance of a component and the reliability associated to each



(a) state equation $x(t)$, μ=5 **(b)** reliability $\rho(t)$, μ=5

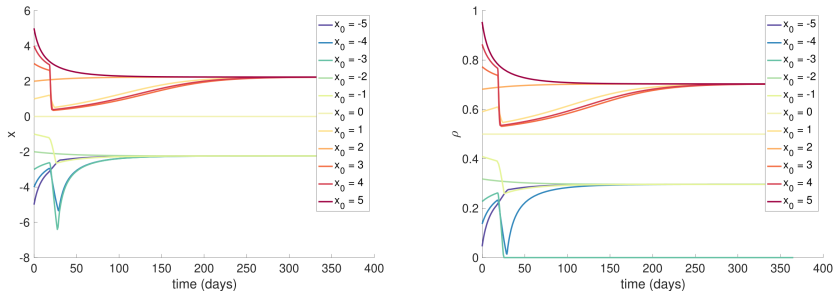
Figure 8.4: Smooth Degradation/Recovery Model. Evolution in time of the state x (a) and the normalised state ρ (b) with different initial points x_0. It is based on the super-critical pitchfork bifurcation in Eq. (8.19) with time-dependent μ following Eq. (8.29).



(a) state equation $x(t)$, μ_0=5 **(b)** reliability $\rho(t)$, μ_0=5

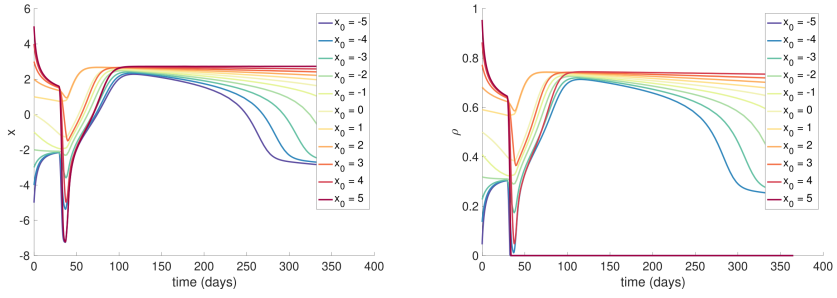
(c) state equation $x(t)$, μ_0=−5 **(d)** reliability $\rho(t)$, μ_0=−5

Figure 8.5: Shock Model. Evolution in time of the state x (a,c) and the normalised state ρ (b,d) with different initial points x_0. The plots are based on the super-critical pitchfork bifurcation in Eq. (8.19) with time-dependent μ in Eq. (8.30).



(a) state equation $x(t)$, μ_0=5 **(b)** reliability $\rho(t)$, μ_0=5

Figure 8.6: Shock and Recovery Model. Evolution in time of the state x (a) and the normalised state ρ (b) with different initial points x_0. The plots are based on the super-critical pitchfork bifurcation in Eq. (8.19) with time-dependent μ following Eq. (8.31).



(a) state equation $x(t)$, μ_0=5 **(b)** reliability $\rho(t)$, μ_0=5

Figure 8.7: Shock and Repair Model. Evolution in time of the state x (a) and the normalised state ρ (b) with different initial points x_0. The plots are based on the bifurcation with hysteresis in Eq. (8.21) with time-dependent μ following Eq. (8.32).

functionality are assumed to depend on a number of uncertain and design system parameters. In the preliminary design phase, this uncertainty is epistemic in nature and thus is here modelled with DST.

The reliability model mixes random occurrences (aleatory) of both *disaster* and *repair* events, during the satellite lifetime, and transitions from fully functional to degraded states (and back) that depend on design solution and epistemic uncertain parameters. The satellite is modelled as a finite multi-state system and the stochastic transitions between states are described as a HCTMC [196]. The reliability model is then integrated into the worst-case scenario optimisation problem by formulating and solving a constrained min-max problem under epistemic uncertainty. Then, an ENM is proposed to represent a complex space system with multiple, coupled subsystems and disciplines. This representation allows one to explore techniques to reduce the computational complexity of evaluating the resilience and robustness of the system.

8.5.1 Optimisation Problem Definition

The goal of the considered optimisation problem is the minimisation of the system mass M_{TOT} under the constraint satisfaction of the expected transmitted data volume f_V. The expected immediate performance $\mathbb{E}\{V(t, X(T); \mathbf{d}, \mathbf{u})\}$ is defined in Eq. (8.84). The uncertainty affects both the probability of transitioning between partial and/or total failure modes and the system mass. With reference to Eq. (8.3), the objective function f corresponds to the total satellite mass M_{TOT} and the constraint function c to the functionality related to the Data Volume transmitted f_V. Problem Eq. (8.3) then translates to:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} M_{TOT}(\mathbf{d}, \mathbf{u}) \\ &s.t. \\ &\nu - \min_{\mathbf{u}\in U} f_V(\mathbf{d}, \mathbf{u}) \leq 0. \end{aligned} \tag{8.33}$$

It has to be noted that the recovery from a partial failure is driven by the value of the design vector $\mathbf{d}$ which, in turns, affects the value of the system mass. The uncertainty domain U is defined by the Cartesian product of the intervals in Table 8.3. In order to facilitate the search for an optimal solution we apply an affine transformation that maps the uncertainty space into a unit hyper-cube where all the uncertainty intervals, along each dimension, are ordered and adjacent [193]. The decision domain D, instead, is defined by the Cartesian product of the intervals in Table 8.2. Where a continuous parameter is used in discrete or binary form, to select a particular component, its value is automatically rounded to the closest integer within the subsystem model.

8.5.2 System Models

It is here presented the mathematical model of the space system, composed of 5 interfaced sub-systems: TTC, OBDH, AOCS, Payload and Power. The ENM framework presented in Section 6.1 has been applied as explained in Section 8.5.4 where the 5 subsystems are translated to nodes and their interconnections to links. A plot of the network is in Fig. 8.8.

Two QoI are here considered: the overall mass of the satellite $M_{TOT}(\mathbf{d},\mathbf{u})$ and the total amount of data sent to the ground station $V(\mathbf{d},\mathbf{u},t)\forall t \in [T_0,T]$:

$$M_{TOT}(\mathbf{d},\mathbf{u}) = \sum_{i=1}^{5} M_i = M_{ttc} + M_{obdh} + M_{aocs} + M_{pl} + M_p \tag{8.34}$$

$$V(\mathbf{d},\mathbf{u},t) = V_i^c + \frac{V_{i+1}^c - V_i^c}{t_{i+1}-t_i}(t-t_i) \quad i = 0,...,N_o-1. \tag{8.35}$$

Both Eqs. (8.34) and (8.35) depend on a vector of decision parameters $\mathbf{d}$ and a vector of epistemic uncertain variables $\mathbf{u}$. Eq. (8.35) is a linear piece-wise interpolation of the components of the vector $\mathbf{V}^c = [V_1^c,...,V_{N_o}^c]^T$ of compressed data volumes sent to the ground station for each of the $\mathbf{T}_o = [T_1,...,T_{N_o}]^T$ periods of the N_o orbits during the total mission time T, such that $T_{i+1} = t_{i+1} - t_i$ and $T = \sum_{i=1}^{N_o} T_i$.

The calculation of the subsystem masses M_{ttc}, M_{obdh}, M_{aocs}, M_{pl}, M_p and of the data volumes V_i^c will be described in more detail in the following sections.

Payload

The payload is a camera that takes images of the Earth during daylight-time T_{dl} and send them to the OBDH sub-system for compression. Since there is not an orbital dynamics node in this example we calculate all the orbital quantities in the payload node. More specifically, the orbit period $T_{orb}(h) = 2\pi\sqrt{\frac{(R_E+h)^3}{\mu}}$, the eclipse time $T_{ecl}(h) = \frac{D_{EA}(h)T_{orb}(h)}{360^\circ}$ and the daylight time $T_{dl}(h) = T_{orb} - T_{ecl}$ [201], that are used by the Payload and the Power nodes, are functions of the uncertain altitude h, where $D_{EA} = 2\arcsin(\frac{R_E}{h+R_E})$ is the Earth Angular Diameter, $R_E = 6.3782\cdot10^3$ km the Earth radius and $\mu_E = 3.986\cdot10^{14}\ m^3s^{-2}$ the Earth gravity constant. The access time to the ground station T_{ac}, that is shared with the TTC node, is defined as:

$$T_{ac} = \frac{T_{orb}}{180^\circ}\arccos\frac{\cos(\zeta_{max})}{\cos(\zeta_{min})} \tag{8.36}$$

where

$$\zeta_{max} = 90^\circ - \epsilon_{min} - \eta_{max} \tag{8.37}$$

$$\sin(\eta_{max}) = \sin\frac{D_{EA}}{2}\cos\epsilon_{min} \tag{8.38}$$

$$\sin(\zeta_{min}) = \sin(L_{pol})\sin(L_{GS}) + \tag{8.39}$$

$$\cos(L_{pol})\cos(L_{GS})\cos(\Delta L) \tag{8.40}$$

with ϵ the elevation angle, η the nadir angle, $L_{pol} = 90^\circ - I$ with I the inclination ($I = I_0 + \delta_{inc}$ with $I_0 = 0$), L_{GS} the latitude at the ground station and ΔL the difference in longitude between orbit pole and ground station [202].

For each completed orbit the payload generates N_i^{pic} images, with $i \in [1, N_o]$. Over several orbits the numbers of images are stored in the vector $\mathbf{N}^{pic}(F_R,h) = [N_1^{pic}, N_2^{pic}, ..., N_{No}^{pic}]^T$, where the number of images per orbit is the product $N_i^{pic} =$

$F_R T_{dl}$ between daylight time and frame rate F_R. The frame rate F_R is evaluated with a piece-wise interpolation of the values $\{6.6, 26.6, 26.6, 26.6\}$ s^{-1} over the design parameter $\tau_{pl} \in \{1, 2, 3, 4\}$. The corresponding amount of data generated by the payload system for each orbit is stored in the vector $\mathbf{V}_{PL}$:

$$\mathbf{V}_{PL} = \frac{I_{mS} B_D \mathbf{N}^{pic}}{2^{33}}, \tag{8.41}$$

which is passed on to the OBDH subsystem. The image size I_{mS} is piece-wise interpolated using the data $\{1280 \times 1024, 640 \times 480, 2592 \times 1944, 1280 \times 1024\}$ pixel, over τ_{pl}. The bit depth B_D is a design parameter and the value at denominator is used to change units from bits to Giga bytes.

Mass and power of the payload are derived from a a look-up table of available cameras. As for the frame rate and image size, by inserting a value of the design parameter τ_{pl}, the model does a piece-wise interpolation returning a mass value from the vector $M_{pl} = [1.1, 1.1, 0.256, 1.1]^T$ kg, a power value in daylight from the vector $P_{pl,dl} = [4, 4, 2.5, 4]^T$ W and a power value in eclipse from the vector $P_{pl,ecl} = [0, 0, 1.75, 9.75]^T$ W [203–205].

On-Board Data Handling

In this system model, it is assumed that the main purpose of the OBDH is to compress and store the images coming from the payload. According to [206], the total compression rate for JPEG compression is C=0.0434. Thus, the volume of data after the compression, that is used in the Eq. (8.35) for the second quantity of interest, is:

$$\mathbf{V}_c = \mathbf{V}_{PL} C. \tag{8.42}$$

The design parameter τ_{obdh} does a piece-wise interpolation of the type of OBDH within a list of four available systems. The model takes the value of τ_{obdh} and linearly interpolates the specific mass and power for the single OBDH module from the vectors: $m^d_{obdh} = [2.3, 2, 1.5, 3]^T$ kg and $p^d_{obdh} = [15, 20, 22, 30]^T$ W. The maximum data storage is $v^d_{obdh} = 4$ Gbytes [207]. The total mass M_{obdh} and the power P_{obdh} of the OBDH are then functions of the compressed data volume $V_c^{max} = \max(\mathbf{V}_c)$, and the uncertain parameters δP_{obdh} and δM_{obdh}:

$$M_{obdh} = m^d_{obdh} \frac{V_c^{max}}{v^d_{obdh}} (1 + \delta M_{obdh}) \tag{8.43}$$

$$P_{obdh} = p^d_{obdh} \frac{V_c^{max}}{v^d_{obdh}} (1 + \delta P_{obdh}) \tag{8.44}$$

Telecommunication System

The TTC is composed of an antenna, an amplified transponder and a Radio Frequency Distribution Network (RFDN). TTC connects the transmitter antenna on the CubeSat with the receiving antenna on the ground station. A patch antenna is considered. The

mass M_{ant} of the antenna depends on the diameter D:

$$D = \frac{\lambda_{ant}}{\pi}\sqrt{\frac{G_t}{\eta_{ant}}} \tag{8.45}$$

with η_{ant} the uncertain antenna efficiency and λ_{ant} the wave length.

$$M_{ant} = \pi\frac{D^2}{4}(0.0005\rho_c + 0.0015\rho_d) \tag{8.46}$$

with $\rho_c = 8940\ kg/m^2$ and $\rho_d = 2000\ kg/m^2$ respectively the density of copper and the density of dielectric material. Eq. (8.46) can be found in [208]. The RFDN mass M_{rfdn} is an uncertain variable while the amplified transponder mass M_{amp} and the power requirement P_{amp} are derived from available data as described in [209], as a function of the transmitter power P_t (power in output from the antenna)

$$P_t = \frac{E_b}{N_0} - G_t - L_t - L_s - L_p - \frac{G_r}{T_{n,s}} + 10\log_{10} R - 228.6 \tag{8.47}$$

and of the amplifier type τ_{amp} (design parameter in Table 8.2). The relations can be found in [209] and are defined from data derived from actual flight hardware. The ratio of received energy-per-bit to noise density, $\frac{E_b}{N_0}$, is a function of frequency f_{ttc}, modulation τ_{mod} and required Bit Error Rate (BER) $= 10^{-5}$ as in [210] where f_{ttc} and τ_{mod} are design parameters. For each modulation type from the list {PSK, BPSK, CFSK, BFSK, FSK, DPSK, QPSK, NRZ} a different formula to evaluate $\frac{E_b}{N_0}$ [210] is given. A linear pairwise interpolation is done of the $\frac{E_b}{N_0}$ values over the τ_{mod} parameter. The quantity L_t is the uncertain on-board loss, while $L_s = 92.44 + 20\log_{10} d_A + 20\log_{10} f_{ttc}$ is the free space path loss with d_A the distance between the transmitter and receiving antennas [210]. The distance d_A is here assumed to be equal to the altitude h for sake of simplicity. The term L_p is the propagation loss and it collects atmospheric attenuation, rain attenuation, pointing loss and other losses that are taken into account in the uncertain parameter L_{other}. $G_r = 60dB$ is the receiver antenna gain. The temperature $T_{n,s}$ is the system noise temperature. $R = \frac{V_c^{max}}{T_{ac}}$ is the data rate, where V_c^{max}, in bits, is the maximum transmitted data volume across all orbits and T_{ac} is the access time to the ground station.

Finally, the mass of the TTC system is the sum of its components:

$$M_{ttc} = M_{ant} + M_{amp} + M_{rfdn}. \tag{8.48}$$

The power of the TTC is a function of the transponder only. In particular, the value in decibel of P_{ttc} is linearly interpolated using the vector $[0.0792, 0.5441]^T$ over the range $[0.1461, 1.9031]^T$ [209]. P_{ttc} is then used as input for the Power subsystem.

Attitude and Orbit Control System

The AOCS is in charge of controlling the orientation of the CubeSat with a three axis stabilisation system. The actuators are reaction wheels and magneto-torquers.

During the mission, the CubeSat is assumed to be affected by a number of disturbances and it is expected to perform some slew manoeuvres. In particular, the solar radiation pressure T_s, the magnetic torque T_m, the torque due to aerodynamic drag T_a and the gravity gradient torque T_g. The torque due to solar radiation pressure is defined as:

$$T_s = l\frac{I_s}{c}A_{sc}(1 + r_f) \tag{8.49}$$

with $I_s = 1420$ W/m^2 the incident solar radiation, c the speed of light, A_{sc} the uncertain area of the surface normal to the sunlight, l the offset between the centre of gravity and centre of pressure of the satellite (a design parameter in Table 8.2) and r_f the uncertain reflectance factor. The torque due to the magnetic field is:

$$T_m = m_{dip}B \tag{8.50}$$

with m_{dip} the uncertain spacecraft residual dipole and B the planet magnetic field strength:

$$B = \frac{B_0 R_E^3}{(R_E + h)^3}\sqrt{3\sin^2(l_M) + 1} \tag{8.51}$$

where l_M is the magnetic latitude. The torque due to drag is defined as:

$$T_a = p_{dyn}C_d A_{sc} l. \tag{8.52}$$

In Eq. (8.52) $p_{dyn} = \frac{1}{2}\rho v^2$ is the the dynamic pressure, where $\rho = \rho_0 e^{-h/H_{sh}}$ is the atmospheric density, with $\rho_0 = 1.2250$ kg/m^3 and $H_{sh} = 8.6$ km, and v the velocity on a circular orbit at altitude h. C_d is the uncertain drag coefficient of the spacecraft. A_{sc} is the uncertain area of the surface normal to the velocity vector considered equal to the surface area in Section 8.5.2 (please refers to Table 8.3 for the value of this uncertain parameter). Note that we assume that both the area of the surface normal to the sunlight and the one normal to the velocity are the same. The torque due to the gravity gradient is:

$$T_g = \frac{3\mu_E}{2(R_E + h)^3}|I_z - \min(I_x, I_y)|\sin 2\psi \tag{8.53}$$

where $I_z = 0.1417(1 + \delta I)$ kg m^2, $I_y = 0.1083$ kg·m^2 and $I_x = 0.0417$ kg·m^2 are the principal moments of inertia of the satellite and $\psi = 8.7266 \cdot 10^{-2}$ radiant is the angle between the spacecraft z-axis and the nadir vector [193]. The total disturbance is the sum:

$$T_d = T_s + T_m + T_a + T_g \tag{8.54}$$

The momentum due to T_d that is stored in the reaction wheels, H_d, and the momentum required for the slew manoeuvres, H_{sl}, are defined as:

$$H_d = \frac{T_d T_{orb}}{4e} \tag{8.55}$$

$$H_{sl} = \frac{4\phi_{sl} I_z}{t_{sl}} \tag{8.56}$$

with $e = 8.7266 \cdot 10^{-2}$ radiant the pointing accuracy, ϕ_{sl} the slew angle and t_{sl} the time allowed for the manoeuvre (design parameters in Table 8.2). The mass, M_{rw}, and power, P_{rw}, of the reaction wheels are computed by interpolation from available real data [209], as functions of the maximum between H_d and H_{sl}:

$$M_{rw} \propto \max(H_d, H_{sl}) \tag{8.57}$$
$$P_{rw} \propto \max(H_d, H_{sl}) \tag{8.58}$$

In particular, for momentums of $[0.0016, 400]^T$ Nms, the masses are respectively $[0.072, 20]^T$ kg and the power consumptions are $[0.465, 110]^T$ W. It is assumed that the momentum stored in the reaction wheels is unloaded with magneto-torquers. The mass and power of the magneto-torquers are interpolated as functions of the required magnetic dipole D_{mag} as in [209]:

$$M_{mt} \propto D_{mag} \tag{8.59}$$
$$P_{mt} \propto D_{mag} \tag{8.60}$$

where

$$D_{mag} = \frac{T_d}{B} \tag{8.61}$$

with B given in Eq. (8.51). In particular, for dipoles D_{mag} of $[0.06, 4000]^T$ Am^2, the masses are respectively $[0.0835, 50]^T$ kg and the power consumptions are $[0.155, 16]^T$ W.
Finally, the outputs of the AOCS node are:

$$M_{aocs} = M_{rw} + M_{mt} \tag{8.62}$$
$$P_{aocs} = P_{rw} + P_{mt} \tag{8.63}$$

Power System

The Electrical Power System (EPS) is composed of a solar array, a battery pack, and a Power Conditioning and Distribution Unit (PCDU). The mass of the power system is the sum of the individual masses of its components

$$M_p = M_{sa} + M_{bp} + M_{pcdu} \tag{8.64}$$

The power produced by the system in daylight is the one generated by the solar array P_{sa}. The design of the solar array is a function of the power requirements during light-time P_{lt} and eclipse P_{ecl} that are calculated from the power requirements of the other subsystems:

$$P_{lt} = 16 + P_{aocs} + P_{ttc} + P_{obdh} + P_{pl,lt}. \tag{8.65}$$
$$P_{ecl} = 16 + P_{aocs} + P_{ttc} + P_{obdh} + P_{pl,ecl}. \tag{8.66}$$

where the number 16 is the base power that accounts for the maintenance of the basic functionalities of the satellite. Given P_{ecl} as well as the duration T_{ecl} of the night, the

energy capacity requirement of the battery system is

$$E_{req} = \frac{P_{ecl}T_{ecl}}{\eta_{b-l}DOD} \tag{8.67}$$

where η_{b-l} is the transfer efficiency between battery and loads and it is the product of the efficiencies of the battery discharge regulator η_{bdr}, the distribution unit η_{du}, and the harness η_{har}:

$$\eta_{b-l} = \eta_{bdr}\eta_{du}\eta_{har} \tag{8.68}$$

The efficiency η_{bdr} of the battery discharge regulator is a function of the bus voltage V_{bus} and is calculated using a linear interpolation of available data [210]. In particular we linearly interpolate the efficiencies [0.90, 0.97] over the voltage range [20, 100] V. The harness efficiency η_{har} is

$$\eta_{har} = 1 - \frac{V_{dr}}{100} \tag{8.69}$$

and is, therefore, dependent on the allowable voltage drop V_{dr} given as a percentage of the bus voltage. The Depth Of Discharge (DOD) is a function of the number $C_L = \frac{T_{tot}}{T_{orb}}$ of charge/discharge cycles, that is dependent on the fixed mission time and on the uncertain altitude h. Their relationship is defined as in [210]:

$$DOD = -36.76 \log \frac{C_L}{207800} \tag{8.70}$$

Given the energy requirement for the battery, the mass of the battery pack is

$$M_{batt} = \frac{E_{req}}{E_c} \tag{8.71}$$

where the energy density E_c (in Wh/kg) is selected from a list of available battery types depending on the capacity $C_B = \frac{E_{req}}{V_{bus}}$. The capacities C_B is used to select the energy density E_c from a look-up table. The model enters with the value C_B to the vector $[1.5, 5.8, 10, 16, 28, 39, 50]^T$ Ah and finds the closest approximation. The corresponding value of the energy density is read from the vector $[115, 133, 139, 155, 118, 126, 165]^T$ Wh/kg [210].

The power P_{sa} required from the solar array is computed considering the duration of the daylight T_{dl}:

$$P_{sa} = \frac{P_{ecl}T_{ecl}}{\eta_{a-b}\eta_{b-l}T_{dl}} + \frac{P_{lt}}{\eta_{a-l}} \tag{8.72}$$

where η_{a-b} is the transfer efficiency between solar array and battery pack, η_{a-l} is the transfer efficiency between solar array and loads. Although the uncertainty on the power requirements comes from all the loads it is assumed that a further epistemic uncertainty exists on the total demand. Therefore an uncertainty factor δP_p is applied to P_{lt} and P_{ecl}: $P_{lt} = P_{lt}(1 + \delta P_p)$ and $P_{ecl} = P_{ecl}(1 + \delta P_p)$. The transfer efficiencies

can be expressed as the product of the efficiencies of the components:

$$\eta_{a-b} = \eta_{sar}\eta_{bcr}\eta_{batt} \tag{8.73}$$

$$\eta_{a-l} = \eta_{sar}\eta_{dist}\eta_{har} \tag{8.74}$$

In Eqs. (8.73) and (8.74)) η_{bcr} is the efficiency of the battery charge regulator and, as for the discharge regulator, it is a function of the bus voltage V_{bus}. Also in this case we interpolate the efficiency [0.90, 0.97] over the voltage range [20, 100] V. The parameter η_{sar} is the efficiency of the solar array regulator, and it is a linear interpolation between 0.94 at 20 V and 0.99 at 100 V when the design parameter τ_{conf} selects the Direct Energy Transfer (DET) configuration, or between 0.93 at 20 V and 0.97 at 100 V when τ_{conf} selects Maximum Power Point Tracker (MPPT) configuration. The efficiency of the distribution unit is $\eta_{dist} = 0.99$. The charging efficiency of the battery is $\eta_{batt} = 0.96$. The array pointing loss factor is

$$\eta_p = \cos\alpha \tag{8.75}$$

where α is the solar incidence angle. The distance r_S, measured in Astronomical Unit (AU), from the Sun involves a loss, or gain, that is

$$\eta_r = \frac{1}{r_S^2} \tag{8.76}$$

Furthermore, cells degrade with time mainly due to radiation fluence, and such degradation can be estimated as:

$$\eta_{life} = (1 - D_c)^T \tag{8.77}$$

where D_c is the cell degradation per year and T is the cell life time (the mission time). A further important factor affecting the efficiency of the solar array is the uncertain assembly efficiency η_a. The efficiency of the array is lower than the efficiency of the single cells because of a loss due to assembly. The total cell efficiency is, therefore, $\eta_{tot} = \eta_a\eta_p\eta_r\eta_{life}$. The specific power (in Wh m^{-2}) of the array is

$$P_{cell} = 1370\eta_c\eta_{tot} \tag{8.78}$$

where η_c is the efficiency of the single solar cell. From this, the required area of the array is computed:

$$A_{sa} = \frac{P_{sa}}{P_{cell}} \tag{8.79}$$

and finally the mass of the solar array

$$M_{sa} = A_{sa}\rho_{sa}. \tag{8.80}$$

The values of D_c, η_c and ρ_{sa} are chosen by the design parameter τ_p. More precisely they are evaluated by a piece-wise interpolation of the following data over the design parameter $\tau_p \in [0,\ 0.5,\ 1]^T$: $\rho_{sa} \in [32 \cdot 10^{-2}, 116 \cdot 10^{-2}, 86 \cdot 10^{-2}]$ kg/m2, $D_c \in [0.0375,\ 0.0275,\ 0.0275]^T$ and $\eta_c \in [0.1555,\ 0.2744,\ 0.2862]^T$. The uncertainty factors δD_c and $\delta\rho_{sa}$ are applied: $D_c = D_c(1 + \delta D_c)$ and $\rho_{sa} = \rho_{sa}(1 + \delta\rho_{sa})$.

The PCDU is a modular unit composed of modules such as battery charge and discharge regulators, solar array regulators, maximum power point tracker, shunt regulator, distribution unit (latching current limiters), telemetry interface. The number of modules, and thus the mass of the unit, depends on τ_{conf}. Indeed, if τ_{conf} is DET, there is no maximum power point tracker, and the PCDU is lighter. On the other hand, an MPPT configuration extracts maximum power from the solar array, therefore the array size decreases, but the presence of the MPPT module decreases the transfer efficiency and increases the PCDU mass. The configuration parameter τ_{conf} is used to trade-off between different components and, thus, is a design parameter. The mass M_{pcdu} can be estimated as the sum

$$M_{pcdu} = \mu_{pcdu}(2P_{sa} + P_{lt} + P_{ecl} + cP_{sa}) \tag{8.81}$$

where $\mu_{pcdu} = 0.001$ kg/W and $c = 0$ for DET and $c = 1$ for MPPT. The factor 2 multiplying the first term in brackets accounts for a telemetry and a distribution unit.

8.5.3 Cube-Sat Resilience Model

We assume that the CubeSat system can be in 3 distinct operational states. State 0: total system failure x_0; state 1: partially functional system x_1; state 2: fully functional system x_2. Each state is associated with a different value of the performance function $V(t, x; \mathbf{d}, \mathbf{u})$.

The assumption underneath the modelling of the resilience of the CubeSat is that a fully, or partially, functional system can deteriorate and a partially functional system can recover but once a total failure of the system occurs the system is not able to recover anymore and the satellite is lost. When the satellite is lost the data volume is zero. At the start of the mission the CubeSat is assumed to be fully functional, which corresponds to a probability of being in state x_2, $P(X(0) = x_2) = 1$. The further assumption is that the occurrence of a complete failure is independent of the occurrences of the partial failures and their recoveries and does not depend on decision and uncertain variables. This is a simplification that will be removed in future developments and does not impair the validity of our results. Thus, following [211], we model the probability of a complete failure of the whole satellite at time t with the Weibull distribution $p_0(t) = \prod_s p_{0,s}(t)$, where $p_{0,s}$ is the Weibull distribution defining the probability of a failure of subsystem s. The individual Weibull density function and associated parameters were taken from [211].

Until a complete failure occurs, the HCTMC in Section 8.3 is used to model the transition between states x_1 and x_2 and back. The stochastic dynamics of this process is given by the transition operator given in Eq. (8.4) with a transition rate matrix

$$Q(\mathbf{d}, \mathbf{u}) = \begin{pmatrix} -\mu & \mu \\ \lambda(\mathbf{d}, \mathbf{u}) & -\lambda(\mathbf{d}, \mathbf{u}) \end{pmatrix}, \tag{8.82}$$

where the first line and column refer to state x_1 and the second ones to state x_2, μ is constant and λ is a function of both design and uncertain parameters. The state of the Cube-Sat changes from x_2 to x_1 with rate λ and with rate μ in the opposite way. A

general solution for the distribution of the system states at any time, conditional upon that the fatal failure has not yet occurred, is given by Eq. (8.5). The simple Markov Chain model we have chosen is well-known within reliability theory as the alternating system with constant rates [78]. Considering our initial conditions ($P(X(0) = x_2) = 1$), conditional on that the fatal failure has not occurred by time t, the probability that the system is in state x_2 at time t can be expressed explicitly as

$$p_2(t) := \Pr(X(t) = x_2 | T_{fail} > t, X(0) = x_2) = \\ = \frac{\mu}{\mu + \lambda} + \frac{\lambda}{\mu + \lambda} \exp(-t(\mu + \lambda)). \quad (8.83)$$

The probability that the system is in state x_1 at time t, conditional upon that the fatal failure has not occurred by time t, will be denoted $p_1(t) = 1 - p_2(t)$. It is the complement of p_2 because of the law of total probability.

The expected value of the instantaneous data increment, which is needed to evaluate the expected total volume of transmitted data Eq. (8.7)), is

$$\mathbb{E}\{V(t, X(T); \mathbf{d}, \mathbf{u})\} = \\ \left[V_2(t; \mathbf{d}, \mathbf{u}) p_2(t) + V_1(t; \mathbf{d}, \mathbf{u}) p_1(t)\right](1 - p_0(t)) + \\ + V_0(t; \mathbf{d}, \mathbf{u}) p_0(t), \quad (8.84)$$

where V_0, V_1 and V_2 represent the instantaneous data increment respectively for states x_0, x_1 and x_2. $V_2(t) = V(t)$ is the data volume for a completely functional satellite. $V_1(t)$ is the data volume of a satellite in the degraded state x_1, and is here computed as:

$$V_1(t) = \frac{V_2(t)}{2} \quad (8.85)$$

When the satellite is in state x_0, total failure, the corresponding data volume is $V_0(t) = 0$.

The parameters μ in Eq. (8.82) is set to the value $1/365$ while parameter λ has a base value $\lambda_0 = 1/365$ and is related to the design and uncertain parameters through the expression

$$\lambda(\mathbf{d}, \mathbf{u}) := \lambda_0 \prod_i \left[r_{u,i}(u_i)\right] \prod_j \left[r_{d,j}(d_j)\right], \quad (8.86)$$

where the two functions $r_{u,i}$ and $r_{d,j}$ represent the relative influence of each of the uncertain or design parameters. This form was chosen because it corresponds to Cox's proportional hazard model [212] with covariates $\mathbf{d}$ and $\mathbf{u}$. If some observations of the process were available, the relative influences could be inferred by statistical methods. In the absence of data, we have chosen an expert estimates for the relations based on linear interpolations between the estimated influences at the lower and upper boundaries of the respective parameter spaces. For $\underline{u}_i, \overline{u}_i$, denoting the lower and upper bound for an uncertain parameter u_i, the respective relative influences at the boundary are denoted $\underline{R}_{u,i}, \overline{R}_{u,i}$ and the relative influence of u_i on the failure transition rate is

$$r_{u,i}(u_i) := \underline{R}_{u,i} + \frac{\overline{R}_{u,i} - \underline{R}_{u,i}}{\overline{u}_i - \underline{u}_i} (u_i - \underline{u}_i) . \tag{8.87}$$

An analogous expression is used to relate $r_{d,i}$ to each d_i. For the sake of the simple exercise presented in this paper, these linear relationships and expression in Eq. (8.86) were purposely constructed to allow the design process to change the rate of transition from x_2 to x_1 in one direction and to allow the uncertain variables to change in the opposite direction. This choice provides a verifiable result. In a more general context, appropriate relationships will need to be defined for each subsystem and component.

We chose the values of $\underline{R}_{u,i}$ and $\overline{R}_{u,i}$ in such a way that each design and uncertain parameter has a different influence on the system degradation and recovery rates.All the values of $\underline{R}_{u,i}$ and $\overline{R}_{u,i}$ are reported in Tables 8.2 and 8.3. The level of influence of each parameter is proportional to $\overline{R}_i - \underline{R}_i$. When this difference is zero, the corresponding parameter is expected to have no effect on the degradation and recovery rates. During the development of the method presented in this chapter, different combinations of parameters and intervals were tested. The particular values reported in Tables 8.2 and 8.3 are only an illustrative example of the many we tested and do not represent any particular system or space mission.

8.5.4 Evidence Network Model and Belief Function Estimation

This section shows how the models previously described are combined to define the ENM framework in Fig. 8.8.

The five sub-systems are translated into 5 network nodes. The two QoI M_{TOT} and $V(t)$ in Eq. (8.33) depend on the 12 design parameters listed in Table 8.2 and the 20 uncertain parameters listed in Table 8.3. Table 8.3 reports the intervals of uncertainty for each parameter with associated bpa in brackets. Part of the bpas was taken from [193] where the authors elicited the opinion of ESA specialists. The remaining were chosen by the author to illustrate the difference between deterministic and resilient solutions.

Table 8.2: Design parameters.

SYSTEMS	**d**	LB	UB	$\underline{R}_d$	$\overline{R}_d$
AOCS	t_{sl} (s)	30	90	1	1
	ϕ_{sl} (deg)	10	60	0.899	1.097
TTC	f_{ttc} (GHz)	7	10	0.85	1.2
	τ_{mod}	0	1	0.95	1.05
	τ_{amp}	0 (TWTA)	1 (SSA)	0.95	1.05
Power	V_{bus} (V)	3	5	0.9	1.1
	V_{dr} (%)	1	5	1	1
	τ_{conf}	0 (DET)	1 (MPPT)	1	1
	τ_p	0	1	1	1
Payload	B_D	1	5	0.9	1.2
	τ_{pl}	1	4	0.9	1.1
OBDH	τ_{obdh}	1	6	0.8	1.2

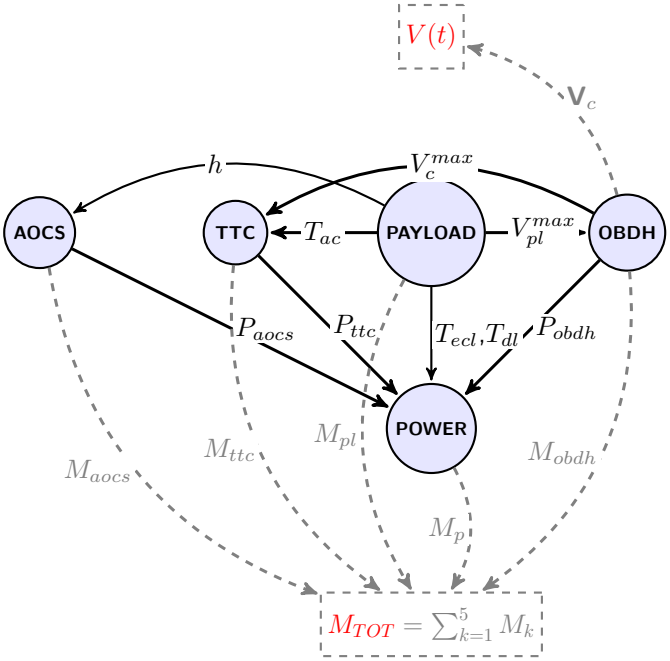


Figure 8.8: Evidence Network Model of the CubeSat. The two quantities of interest are the mass of the CubeSat M_{TOT} and the total amount of data transmitted to the ground station V; M_{TOT} is the sum of the mass of the 5 subsystems and V is the quantity of data sent by the TTC after the compression in OBDH.

Table 8.3: Uncertain parameters.

Systems	**u**	interval 1 (*bpa*)	interval 2 (*bpa*)	$\underline{R}_u$	$\overline{R}_u$
Payload	h (km)	[600 800] (0.4)	[800 1000] (0.6)	0.9	0.967
	ϵ (deg)	[0 5] (0.4)	[5 10] (0.6)	1	1
	δ_{inc} (*deg*)	[0 5] (0.3)	[5 10] (0.7)	1	1
OBDH	δP_{obdh}	[0 0.1] (0.5)	[0.1 0.2] (0.5)	1	1
	δM_{obdh}	[0 0.1] (0.8)	[0.1 0.2] (0.2)	1	1
AOCS	l (m)	[0.005 0.01] (0.5)	[0.01 0.02] (0.5)	0.94	1.2
	A_{sc} (m^2)	[0.034 0.0885] (0.5)	[0.0885 0.15] (0.5)	1	1
	r_f	[0.5 0.6] (0.5)	[0.6 0.7] (0.5)	1	1
	m_{dip} (mA $\cdot m^2$)	[0.5 1] (0.5)	[1 1.5] (0.5)	0.85	0.98
	C_D	[2 2.2] (0.4)	[2.2 2.5] (0.6)	0.9	1.1
	δI	[-0.1 0.05] (0.5)	[0.05 0.1] (0.5)	0.85	1
TTC	η_{ant}	[0.6 0.8] (0.3)	[0.8 0.9] (0.7)	1	1
	G_t (dB)	[1 3] (0.3)	3 5 (0.7)	1	1.15
	L_t (dB)	[0.1 0.5] (0.3)	[0.5 1] (0.7)	1	1.05
	L_{other} (dB)	[0.5 1.5] (0.4)	[1.5 2.0] (0.6)	0.85	1
	M_{rfdn} (kg)	[0.1 0.3] (0.4)	[0.2 0.5] (0.6)	1	1
Power	δD_c	[0.025 0.0275](0.4)	[0.3 0.0375] (0.6)	1	1
	η_a	[0.8 0.85] (0.4)	[0.85 0.9] (0.6)	0.8	1
	$\delta\rho_{sa}$ (kg/m^2)	[3.5 3.6] (0.3)	[3.6 4] (0.7)	1	1
	δP_p	[0 0.1] (0.5)	[0.1 0.2] (0.5)	0.95	1.05

The ENM is built to model only the influence of the uncertain parameters. This influence is transmitted via a scalar positive quantity. Hence all solid links in Fig. 8.8 represent the propagation of the effect of the most influential uncertain parameters. Indeed, after a preliminary sensitivity analysis, the dependency between Payload and TTC through δ_{inc} and ϵ was found to be poorly influential. As a reasonable approximation, it has been removed from the link. In the figure, the dashed lines indicate the contributions of all the subsystems to the total system mass and the total data volume. Given the ENM in Fig. 8.8, the uncertain vector $\mathbf{u}$ can be partitioned into the uncoupled vector:

$$\mathbf{u}_u = [\delta M_{obdh}, r_f, m_{dip}, \eta_{ant}, M_{rfdn}, \delta D_c, \eta_a, \delta\rho_{sa}, \delta P_p]^T \tag{8.88}$$

and the coupled vector:

$$\mathbf{u}_c = [l, A_{sc}, C_D, \delta I, G_t, L_t, L_{other}, \delta P_{obdh}, h, \epsilon, \delta_{inc}]^T. \tag{8.89}$$

Once the uncertain parameters are partitioned into coupled and uncoupled, one can write the total mass as an explicit function of the two groups of parameters and of the scalar exchange functions φ_{ij} (namely scalar quantities V_c^{max}, T_{ac}, V_{pl}^{max}, P_{aocs}, P_{ttc}, P_{obdh}, T_{ecl}, T_{dl} and h as represented in Fig. 8.8):

$$\begin{aligned} &M_{TOT} = M_{aocs}(h, r_f, m_{dip}, l, A_{sc}, C_D, \delta I)+ \\ &M_{ttc}(V_c^{max}(h), T_{ac}(h), \eta_{ant}, M_{rfdn}, G_t, L_t, L_{other})+ \\ &M_{pl} + M_{obdh}(V_{pl}^{max}(h), \delta M_{obdh})+ \\ &M_p(P_{aocs}(h, l, A_{sc}, C_D, \delta I), P_{ttc}(V_c^{max}(h), \\ &T_{ac}(h), G_t, L_t, L_{other}), P_{obdh}(V_c^{max}(h), \delta P_{obdh}), \\ &T_{ecl}(h), T_{dl}(h), \delta D_c, \eta_a, \delta\rho_{sa}, \delta P_p) \end{aligned} \tag{8.90}$$

where only the dependencies on the uncertain parameters are made explicit. Note that M_{pl} does not depend on any uncertain parameter and that the values of δ_{inc} and ϵ in the calculation of the access time were fixed to the value coming from the worst case analysis, due to their low influence on the calculation of mass and power. Furthermore, five exchange functions, T_{ac}, T_{ecl}, T_{dl}, V_{pl}^{max} and V_c^{max}, all depend on the same uncertain parameter h, hence in the following all these links will be treated as one and a partial belief curve will be computed for the overall influence of h on the calculation of the system mass.

With this ENM and related partitioning of the uncertain vector, one can apply the decomposition proposed in Algorithm 11 and generate a lower estimation of the Bel with a total cost of $28 + 26N_s$ optimisations, where the parameter N_s is the number of FEs samples from the partial curves (see Eq. (7.13)). In comparison, an exact calculation of the Bel would require a total of $N_{FE}^{full} = 2^{20} = 1048576$ optimisations.

The number of influential links that we propose for the construction of the specific ENM in Fig. 8.8 only serves the scope to develop an exercise that proves the effectiveness of the methodology we described in previous sections. More complex and realistic interactions among subsystems are clearly possible but do not imply a modification of the method. They would simply scale the computational complexity as in Eq. (7.13).

8.5.5 Results

The computer used for the simulations is a Microsoft Windows 10 Pro, x64-based, Intel(R) Core(TM) i7-6700 CPU, 3.40 GHz, 3408 MHz, 4 cores, 8 Logical Processors, 8 GB (RAM) and the software is implemented in MATLAB R2018b.

Consider first the min-max problem in Eq. (8.33). The results are represented in Fig. 8.9 for 4 different values of the threshold ν (represented by a vertical line): 500, 600, 700 and 800. For each ν the figure shows the optimal mass that corresponds to the robust design vector $\mathbf{d}_{minmax}$, which satisfies the reliability constraint in Eq. (8.33) for all values in the uncertain domain U.

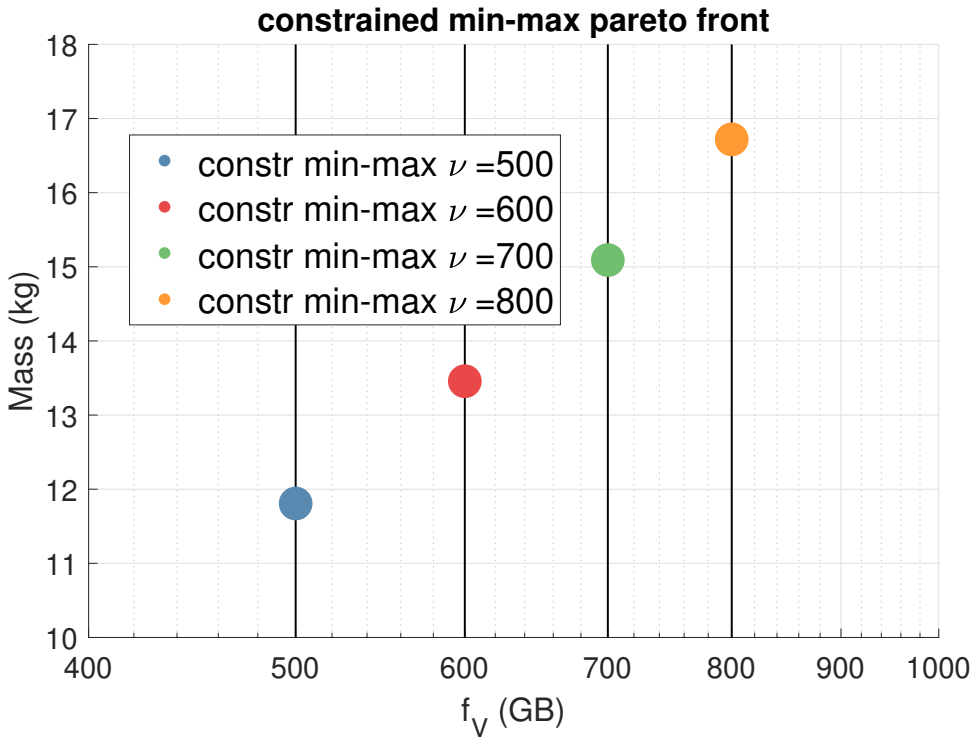


Figure 8.9: Results for the constrained min-max optimisation: each point represents the minimum worst-case value in the uncertain space for both objective and constraint functions. It is the projection of the two worst cases as in Fig. 8.10.

Fig. 8.10 further explains the results in Fig. 8.9. Each optimal solution is represented now by two points and a line that connects them. The two points correspond to the same design solution $\mathbf{d}_{minmax}$ but to two different uncertain vectors $\mathbf{u}$. The circle is the worst-case value for the mass M_{TOT}

$$\mathbf{u}_{\max,M} = \arg\max_{u \in U} M_{TOT}(\mathbf{d}, \mathbf{u}) \tag{8.91}$$

and the diamond is the worst-case value for the constraint

$$\mathbf{u}_{\max,V} = \arg\max_{u \in U} \big(\nu - f_V(\mathbf{d}, \mathbf{u})\big). \tag{8.92}$$

In all four cases, the maximum constraint violation is equal to zero, thus all decision vectors $\mathbf{d}$ are always feasible. This figure also shows that the mass is maximised for a $\mathbf{u}$ vector that is inside the feasible domain.

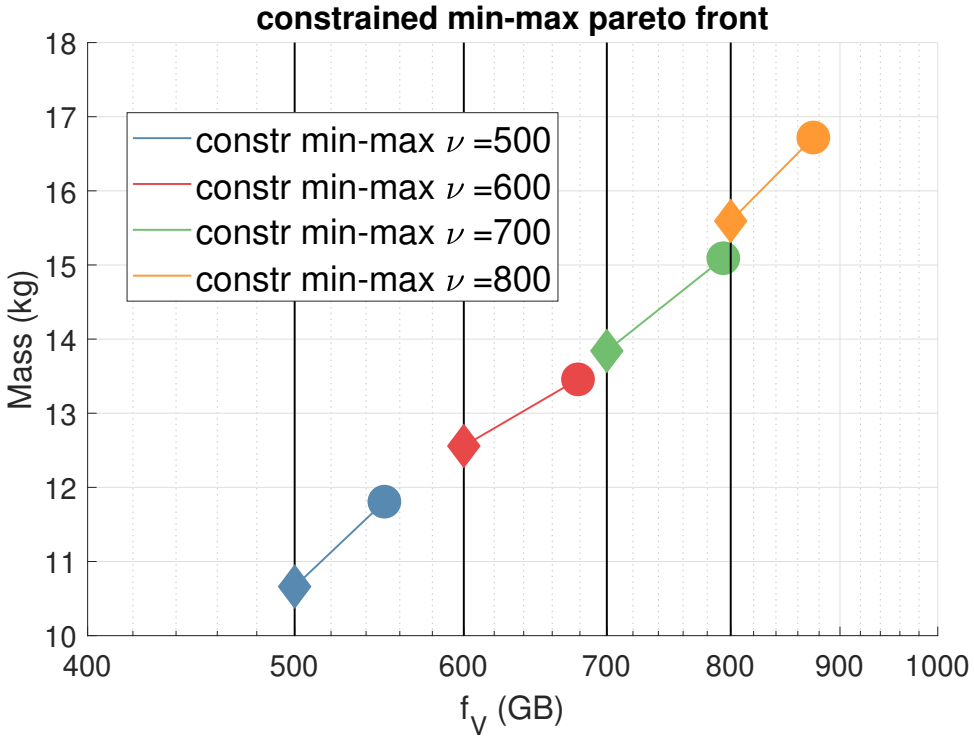


Figure 8.10: Results for the constrained min-max optimisation. Each optimal solution is represented by two points and a line that connects them. The two points correspond to the same design solution $\mathbf{d}_{minmax}$ but to two different uncertain vectors $\mathbf{u}$. The circle is the worst-case value for the mass M_{TOT} and the diamond is the worst-case value for the constraint f_V. The projection of the circle on the corresponding vertical line produces the summary plot in Fig. 8.9.

Fig. 8.11 compares a particular solution from Fig. 8.9 (the one with $\nu = E(V) = 600$) with the solution of the following deterministic optimisation problem, where the uncertain vector $\mathbf{u}$ was set to the value $\mathbf{u}_{nom}$ (the mean value of the intervals defined in Table Table 8.3):

$$\begin{aligned} &\min_{\mathbf{d} \in D} M_{TOT}(\mathbf{d}, \mathbf{u}_{nom}) \\ &s.t. \\ &\nu - f_V(\mathbf{d}, \mathbf{u}_{nom}) \leq 0. \end{aligned} \tag{8.93}$$

The red point is the optimal resilient solution $(\mathbf{d}_{minmax}, \mathbf{u}_{minmax})$ calculated with the EBORe approach proposed in this paper, where $\mathbf{d}_{minmax}$ is in Table 8.4 and $\mathbf{u}_{minmax}$

$= [2.0000{\cdot}10^{-2},\ 1.5000{\cdot}10^{-1},\ 7.0000{\cdot}10^{-1},\ 1.5000{\cdot}10^{-3},\ 2.3447,\ 2.6608,\ 6.0000{\cdot}10^{-1},\ 1.0000,\ 1.0000,\ 2.0000,\ 5.0000{\cdot}10^{-1},\ 8.0000{\cdot}10^{-1}\ 2.0000{\cdot}10^{1},\ 3.0000{\cdot}10^{1},\ 1.0000{\cdot}10^{2},\ 1.0000{\cdot}10^{3},\ 1.0000{\cdot}10^{1},\ 1.0000{\cdot}10^{1},\ 2.0000{\cdot}10^{1},\ 2.0000{\cdot}10^{1}]^T$. The blue square is the solution to the problem in Eq. (8.93); the green hexagram is the worst possible mass due to uncertainty, given the solution of the problem in Eq. (8.93), $\mathbf{d}_{nom}^{opt}$; the yellow pentagram is the minimum value of f_V due to uncertainty, given the design vector $\mathbf{d}_{nom}^{opt}$. From this figure, one can see that by not accounting for the full variability of the uncertain parameters, problem Eq. (8.93) returns a solution that has a lower mass than the resilient one but violates the constraint on the data volume for some values of the uncertain parameters (yellow pentagram) and produces a worst-case mass increase that also violates the constraint on the data volume (green hexagram).

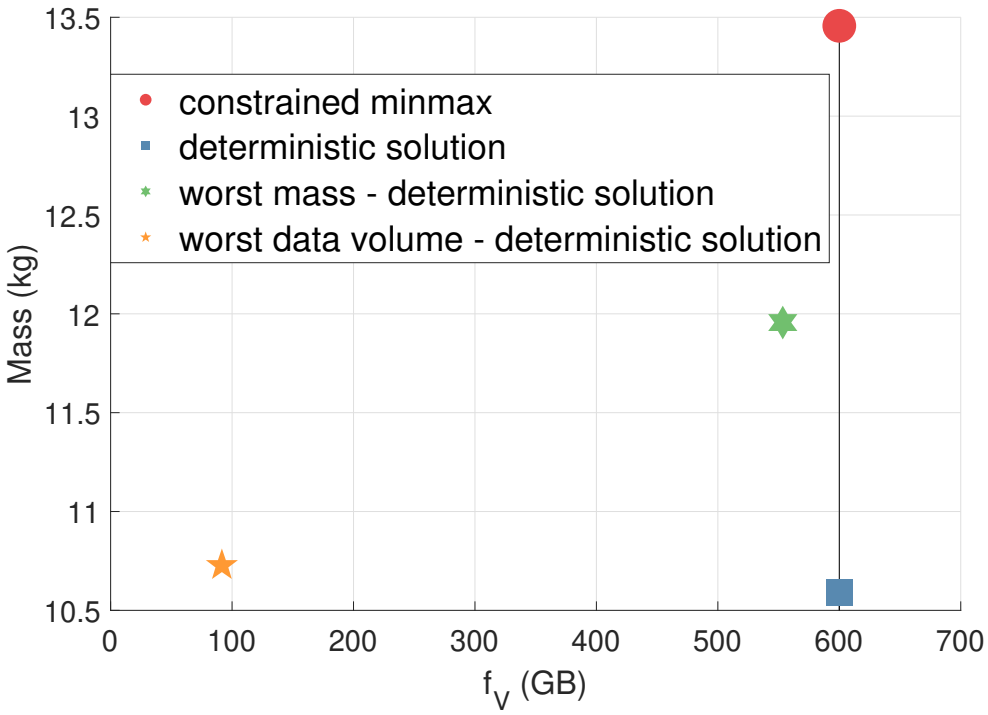


Figure 8.11: Comparison, with $\nu = 600$, of constrained and unconstrained min-max and deterministic approach.

Fig. 8.12 shows the Belief curve for the solution corresponding to $\nu = 600$. It shows in particular, the convergence of the algorithm for the quantification of Belief. Two samples from each partial Bel curve correspond to a total of $N_s = 2^4 = 16$ samples and $N_{FE}^{Dec} = 450$ optimisations. With a maximum time-cost of 10^{-3}s for each function evaluation (because each subsystem function is called individually), each full belief curve requires 7 minutes. It is worth reminding at this point that the decomposition approach used to reconstruct the belief curves starts from the solution of the min-max problem which is assumed to have $Bel = 1$. The reconstruction of the curves confirms the correctness of the min-max as no worse solution in the U space is found. Note

also that a full exact reconstruction of the belief curves would require 2^{20} optimisations against the 450 required with the decomposition.

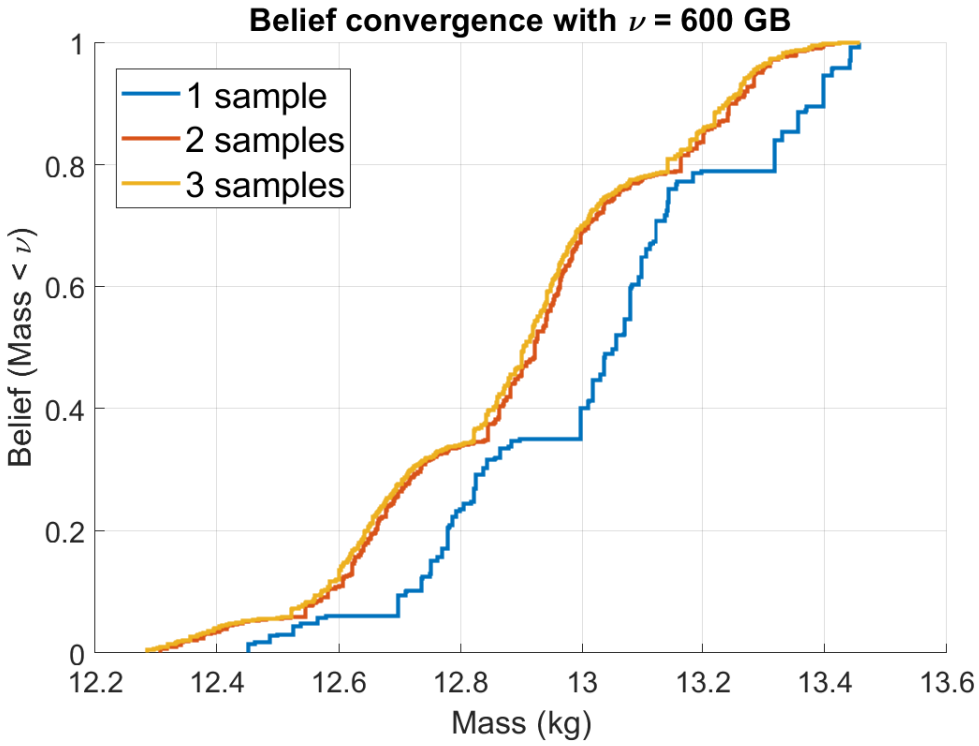


Figure 8.12: Convergence of the belief curves calculated with the decomposition approach.

In Fig. 8.13 we compare the resilient solution $\mathbf{d}_{minmax}$ corresponding to $f_V = 600$ from Fig. 8.9 with a non-resilient solution $\mathbf{d}_{archive} = [1.0007{\cdot}10^{1},\ 4.8123{\cdot}10^{1},\ 9.7875,\ 1.4981{\cdot}10^{-4},\ 4.0505{\cdot}10^{-1},\ 9.9803{\cdot}10^{-1},\ 4.7210,\ 2.4052{\cdot}10^{-1},\ 1.1660,\ 1.0057,\ 2.5439{\cdot}10^{-1},\ 7.3898{\cdot}10^{-1}]^T$ that is feasible in all the uncertain space U. The resilient solution corresponds to the dotted Bel curve in blue, while the non-resilient solution, with $\mathbf{u} = \mathbf{u}_{nom}$, corresponds to the dashed vertical line. Following the normal practice [54] and considering the satellite as an item to be developed, a 20% margin was added to each subsystem mass of the non-resilient solution. Also, a 20% margin was added to the power requirements of the TTC, OBDH, AOCS and Payload subsystems. The non-resilient solution plus margins is the solid vertical line.

One can then build the Bel curve also for the non-resilient solution (dotted line in Fig. 8.13). From this simple comparison one can see that the non-resilient solution without margins has $Bel = 0$ to be realised. The one with margins does not achieve $Bel = 1$ but only $Bel = 0.05$ to be realised and is oversized compared to the resilient solution. Although the non-resilient solution in this example is arbitrary, the result demonstrates that an improper quantification of uncertainty can lead to an undesirable design solution even if the recommended subsystem and system level margins are used.

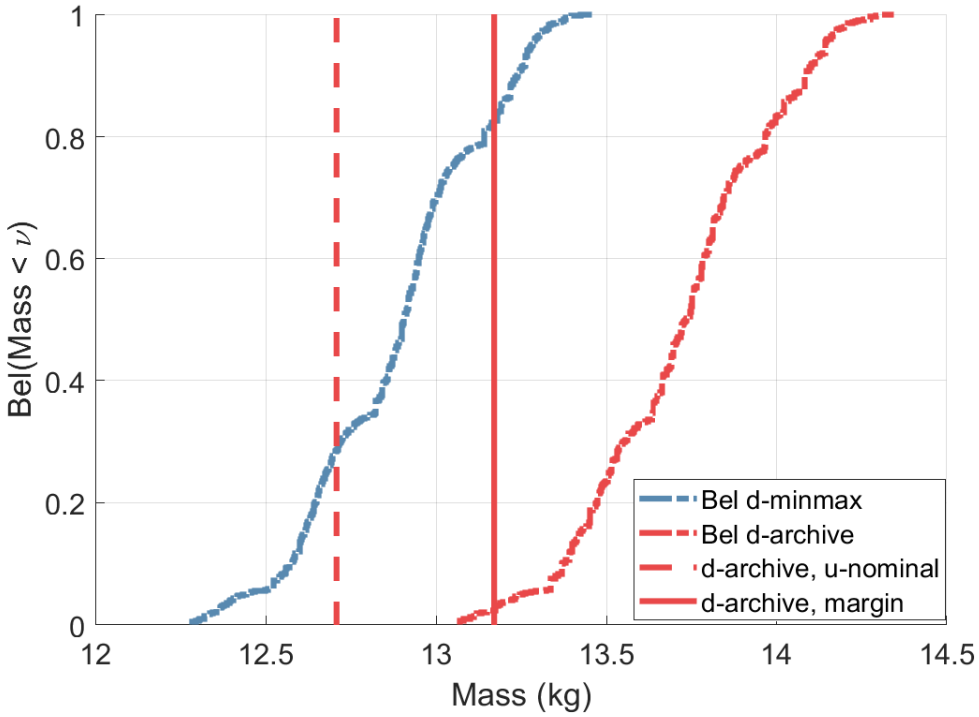


Figure 8.13: The plot shows two comparisons. First, the nominal solution (dashed vertical red line) and the nominal plus margin (vertical red line) are compared with the propagated uncertainty in the same design solution (red belief curve). Then, the nominal solution (red lines) is compared with the resilient solution (blue belief curve).

Fig. 8.14 shows the Belief surface that corresponds to the condition:

$$Bel(M_{TOT} < \nu_M \ \wedge \ f_V > \nu_V) \tag{8.94}$$

where the two thresholds ν_M and ν_V are assumed to be independent from each other. While the cumulative belief distribution in Fig. 8.13, blue dotted line, represents the effect of uncertainty on the system mass M_{TOT} for $f_V = 600$, one could be interested in the belief that both (M_{TOT} and f_V) satisfy condition in Eq. (8.94) at the same time. The resulting Belief-surface in Fig. 8.14 extends the Belief-curve in Fig. 8.13 by adding the evidence in support of the achievement of the values of f_V. By sectioning the surface with cuts parallel to the axes one can find, for any fixed value of f_V or M_{TOT}, the corresponding Belief-curve ($Bel(f_V > \nu_V)$ or $Bel(M_{TOT} < \nu_M)$. Fig. 8.14 shows that, in order to have a joint $Bel > 0.8$ that both expected data volume and mass are correct, one needs to assume a mass larger than 12.9 kg and a data volume lower than 620 GBit. However, it has to be noted that the Belief values on the expected data volume were computed still using the ENM in Fig. 8.17. Thus one has to interpret the result in Fig. 8.14 as the evidence in support to the expected data volumes associated to the values of the mass that can be computed with the ENM.

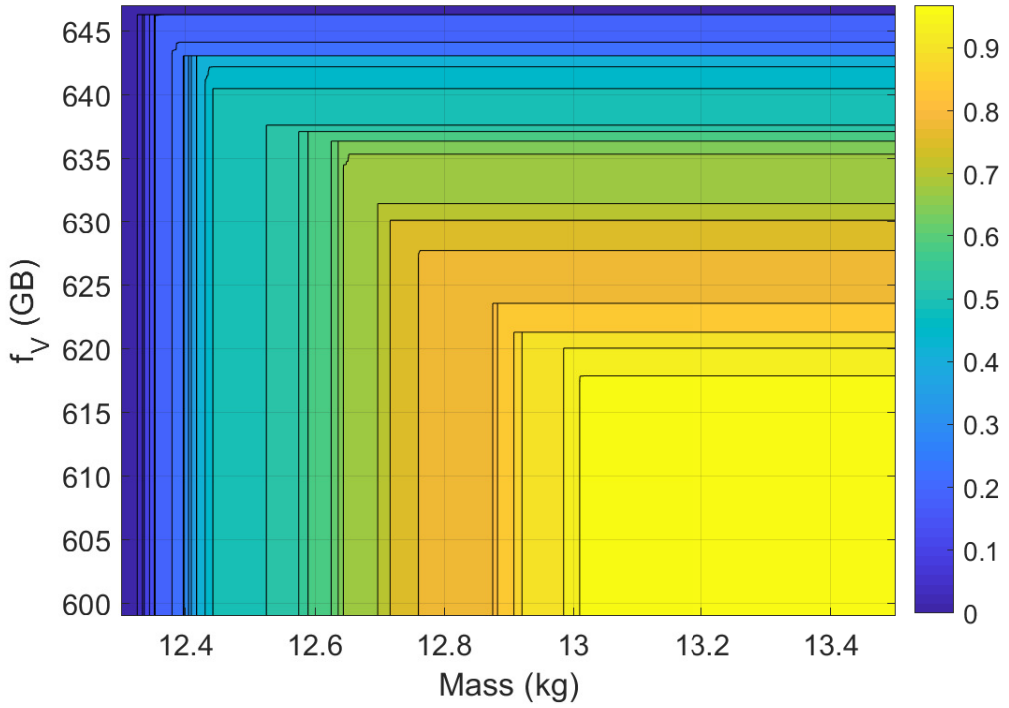


Figure 8.14: Belief surface for the constrained problem formulation with the design vector $\mathbf{d}_{minmax}$. Both mass M_{TOT} and expected data volume f_V are considered.

In Fig. 8.15 and Fig. 8.16, finally, only the constraint function f_V is considered. Five deterministic solutions, including the optimal-deterministic solution with $\mathbf{u}_{nom}$, and the resilient solution $[\mathbf{d}_{minmax}, \mathbf{u}_{minmax}]$ with the constraint $f_V > 600$ are compared. Table 8.4 lists the design vectors. The histograms show the normalised results for 10000 simulations where the time span covered by each mission is 365 days. In particular, Fig. 8.15 compares the total number of transitions from one state (0, 1 and 2) to another while Fig. 8.16 shows the cumulative time spent in each spacecraft state divided by 365 times the number of simulations.

The comparison proves that the resilient design solution increases the probability of the whole system of being in the fully functional state x_2 and decreases the number of transitions from state x_2 to the partial functioning state x_1. It also shows that the resilient solution is always the best in terms of time spent in state x_2. On the contrary, a random design solution may lead to a much longer time spent in the partially functioning state x_1. Note that all bars in the histogram correspond to the worst uncertainty vector for the expected data volume.

The optimal deterministic solution was computed using 50000 function evaluations, compared to the 200000 used to compute the resilient solution. However, the higher computational cost of the min-max solution is repaid by a lower failure rate as shown in Fig. 8.15 and Fig. 8.16. More importantly, Fig. 8.11 has shown that the effect

Table 8.4: Design vectors of Fig. 8.15 and Fig. 8.16

parameter	design 1	design 2	design 3	design 4	d_{minmax}	d_{opt}
t_{sl} (s)	41.730	10.000	10.081	10.000	10.000	10.000
ϕ_{sl} (deg)	41.954	50.685	78.239	52.453	53.631	75.157
f_{ttc} (GHz)	8.413	10.000	9.946	10.000	10.000	10.000
τ_{mod}	0.602	1.000	0.331	1.000	0.333	0.333
τ_{amp}	0.049	0.500	0.500	0.500	0.499	0.500
V_{bus} (V)	0.400	0.000	0.000	0.000	0.000	0.000
V_{dr} (%)	3.026	5.000	4.307	5.000	5.000	5.000
τ_{conf}	0.374	0.413	0.146	0.486	0.201	0.278
τ_p	2.380	1.000	1.069	1.000	1.000	1.000
B_D	3.837	1.000	1.075	1.000	1.000	1.000
τ_{pl}	0.852	0.343	0.045	0.259	0.061	0.022
τ_{obdh}	0.815	0.750	0.750	0.750	0.749	0.750

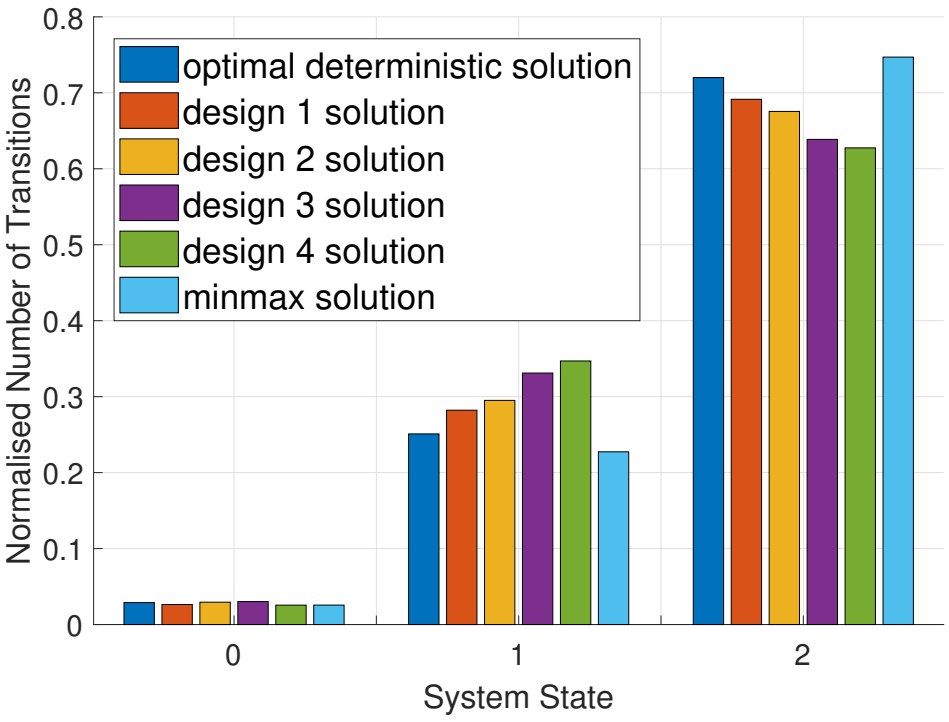


Figure 8.15: Comparison of five deterministic design solutions and the resilient solution (minmax) over the number of transitions between the three system's states (0,1,2).

of uncertainty leads to a considerable increase in mass with respect to the min-max solution and a substantial violation of the reliability constraint.

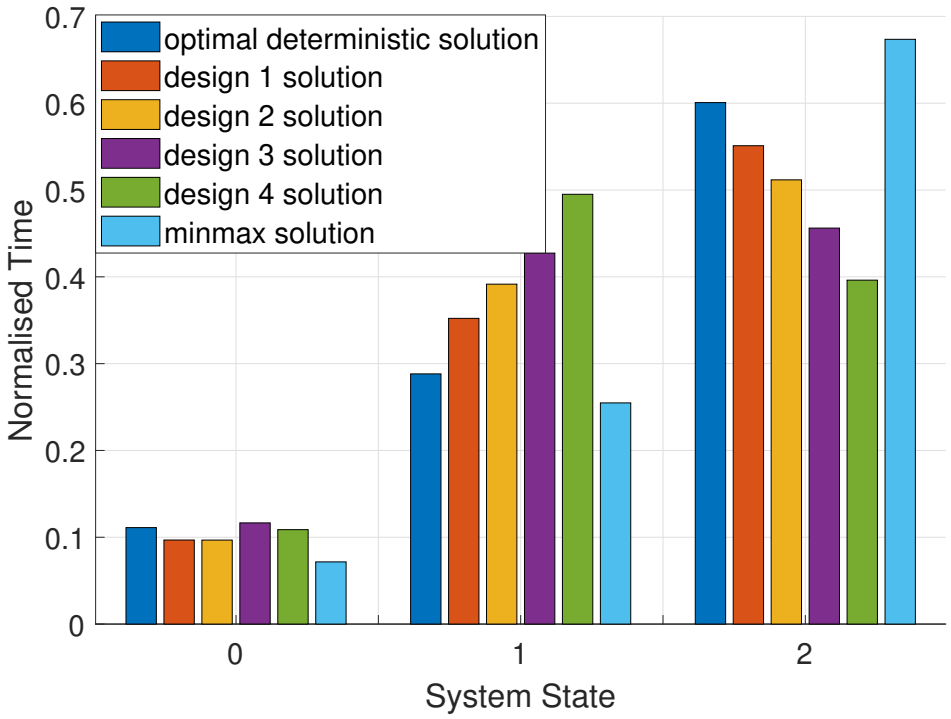


Figure 8.16: Comparison of five deterministic design solutions and the resilient solution (minmax) over the time spent in each system's state (0,1,2).

8.6 Test Case Application (Catastrophe Theory Model)

A similar test problem to the one presented in Section 8.5 is here used for the EBORe approach with Catastrophe Theory for the resilience model.

The system under design is a satellite in LEO. The goal is to take pictures of the Earth's surface and transmit them to a ground-based receiver. The satellite is assumed to be composed of 6 subsystems: AOCS, TTC, OBDH, Power, Orbit and Payload subsystems. The space system is modelled as a network: each subsystem corresponds to a node and each dependency to a link. The ENM is represented in Fig. 8.17.

The design $\mathbf{d}$ and uncertain $\mathbf{u}$ variables are finally listed in Tables 8.5 and 8.6 together with the corresponding sub-systems.

8.6.1 Optimisation Problem Definition

We apply here the general EBORe problem formulation presented in Eq. (8.3):

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} M_{TOT}(\mathbf{d},\mathbf{u}) \\ &s.t. \\ &\nu_i - \min_{\mathbf{u}\in U} f_V(\mathbf{d},\mathbf{u}) \leq 0 \quad \forall i \in I_\nu = [1,...,s]^T \end{aligned} \tag{8.95}$$

Table 8.5: Spacecraft model - design parameters

design parameter	N	sub-system
width for square detector	d_1	Payload
quality factor for imaging	d_2	Payload
operating wavelength	d_3	Payload
obdh type	d_4	OBDH
compression factor	d_5	OBDH
slew angle	d_6	AOCS
time for slew manuvers	d_7	AOCS
frequency	d_8	TTC
modulation	d_9	TTC
amplifier type	d_{10}	TTC
cell type	d_{11}	Power
bus voltage	d_{12}	Power
allowed bus drop	d_{13}	Power

Table 8.6: Spacecraft model - uncertain parameters

uncertain parameter	N	sub-system
altitude	u_1	Orbit
elevation angle	u_2	Orbit
inclination	u_3	Orbit
maximum incidence angle	u_4	Payload
max ground sampling distance	u_5	Payload
Δ mass	u_6	OBDH
Δ power	u_7	OBDH
antenna efficincy	u_8	TTC
antenna gain	u_9	TTC
mass distribution network	u_{10}	TTC
cell packing efficiency	u_{11}	Power
harness mass factor	u_{12}	Power
worst case angle of incidence	u_{13}	Power

where M_{TOT} is the total mass of the satellite and f_V is the volume of data transmitted to the ground station. A set of thresholds ν_i, will be used in the following in order to reconstruct the trade-off between performance $f = M_{TOT}$ and functionality $g = f_V$. This can be considered as CMOP approach to min-max where a ECS is used.

In Eq. (8.95), the objective function is the time-independent mass M_{TOT}

$$f \coloneqq M_{TOT}(\mathbf{d}, \mathbf{u}) = \sum_{i=1}^{6} mass_i(\mathbf{d}_i, \mathbf{u}_i, h_i(\mathbf{d}_{ij}, \mathbf{u}_{ij})) \tag{8.96}$$

where h_i is the set of coupling functions between node i and all the nodes that are linked to it. The quantity f_V is calculated instead integrating over time the product of the compressed data volume DV^c and system state function ρ:

$$g \coloneqq f_V(\mathbf{d}, \mathbf{u}, t) = \int_{T_0}^{T_M} DV^c(\mathbf{d}, \mathbf{u}, t)\rho(\mathbf{d}, \mathbf{u}, t)dt. \tag{8.97}$$

where $DV^c(\mathbf{d}, \mathbf{u}, t)$ represents the QoI and $\rho(\mathbf{d}, \mathbf{u}, t)$ is the global reliability model obtained by normalisation of the dynamics $x(\mathbf{d}, \mathbf{u}, t)$ as explained in Section 8.4.

Following Eq. (8.95), we want to optimise the satellite considering the worst case in the uncertainty for both performance and functionality. In other terms, we want to minimise the satellite mass while ensuring a minimum amount of compressed data volume for any possible scenario in the uncertain space. This could bring to a penalisation of the QoI if it means a significant increase for the reliability ρ, or vice versa. Also, a design configuration that could bring shocks in the resilience function ρ, can be an optimal solution if it guarantees a recovery and a good functionality state after the shock.

For the sake of simplicity and clarity, we study only the dynamics of the node OBDH $x = x_\mu^{OBDH}(\mathbf{d}, \mathbf{u}, t)$. The function x describes the process of continuous degradation and recovery of the system that we are optimising within the EBORe approach. Furthermore, we consider x as a function of two variables: the compression factor $d_5 \in [0.2, 0.6]$ (design) and of the **Δmass** $u_6 \in [0, 20]$ (uncertain): $x(d_5, u_6, t)$. In particular, the functions μ and x_0 depend on d_5 and u_6 as explained in the following. The global reliability model $\rho(d_5, u_6, t)$ is then obtained by normalisation of x as described in Section 8.4.2. The following set of problems has been solved:

- unconstrained SOP min-max
- CMOP min-max with super-critical Pitchfork bifurcation, $\mu(d_5)$ and $x_0(u_6)$;
- CMOP min-max with super-critical Pitchfork bifurcation, $\mu(u_6)$ and $x_0(d_5)$;
- CMOP min-max with sub-critical Pitchfork bifurcation, $\mu(d_5)$, $x_0(u_6)$
- CMOP min-max with shock and recovery on the hysteresis bifurcation, $\mu(d_5, t)$ and $x_0(u_6)$.

The parameter μ and the initial value x_0 in the following subsections have been defined to have a trade-off between f and g.

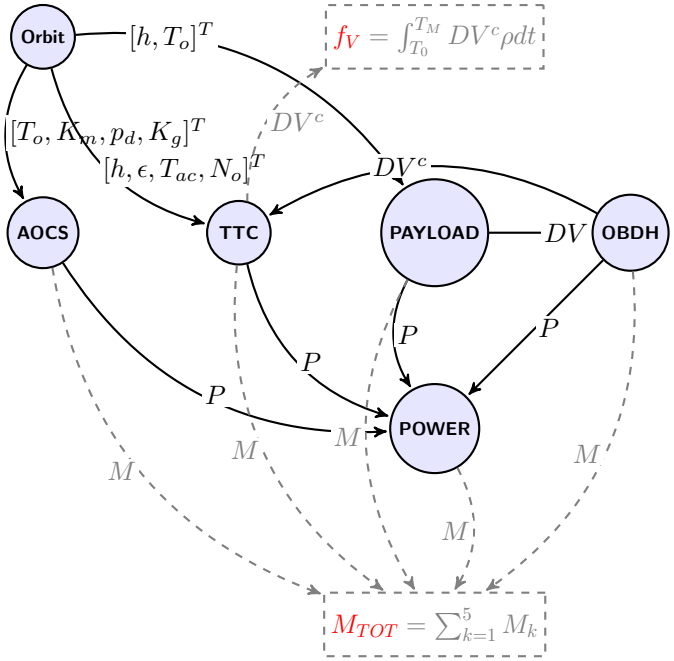


Figure 8.17: Representation of the spacecraft as a complex system. The two quantities of interest are $mass$ and the total amount of data compressed by the OBDH sub-system DV^c.

8.6.2 Results

This subsection presents the results of the EBORe approach to the design for resilience of a satellite in LEO for Earth imaging with problem formulation in Eq. (8.95) and bifurcation theory used to model global reliability. Results are reported based on the classification explained in the previous section.

Unconstrained Problem

The unconstrained min-max problem

$$\min_{d \in D} \max_{u \in U} Mass(\mathbf{d}, \mathbf{u}) \tag{8.98}$$

is first solved. The solution of Eq. (8.98) gives $d_5 = 0.2$ with $d_5 \in [0.2, 0.6]$ and $u_6 = 20$ with $u_6 \in [0, 20]$. This will be important in the following to compare the results of the different problems.

Super-Critical Pitchfork Bifurcation, $\mu(d_5)$, $x_0(u_6)$

The super-critical pitchfork bifurcation model in Eq. (8.19) is used here for the dynamics x (and then for the global reliability function ρ). In particular, we consider the parameter $\mu \propto d_5$ to be a design variable and the initial condition $x_0 \propto u_6$ to be uncertain:

$$\begin{aligned} \mu(d_5) &= -12.5d_5 + 7.5 \\ x_0(u_6) &= 0.5u_6 - 5 \end{aligned} \tag{8.99}$$

In Eq. (8.99) the parameter μ interpolates the values $\{5,\ 0\}$ over the design parameter $d_5 \in [0.2,\ 0.6]^T$. The parameter x_0 interpolates the values $\{-5,\ 5\}$ over the uncertain parameter $u_6 \in [0,\ 20]^T$. In this way, the model has been set such that $\mu(d_5 = 0.2) = 5$, is the best for the mass and the worst for the reliability function. On the other side, $\mu(d_5 = 0.6) = 0$ becomes optimal for ρ but the worst for the mass.

Following the Eq. (8.19), the dynamics x becomes:

$$\begin{aligned} \dot{x} &= \mu(d_5)x - x^3 \\ x_0 &= x_0(u_6) \end{aligned} \tag{8.100}$$

The global reliability $\rho(\mathbf{d}, \mathbf{u}, t)$ is calculated from the solution x of Eq. (8.100) through the normalisation process explained in Section 8.4.2.

The function $\rho(\mathbf{d}, \mathbf{u}, t)$ is finally used to define the functionality $g = f_V$ in Eq. (8.97) that represents the constraint in the Optimisation Problem Formulation in Eq. (8.95). In all the scenarios, the worst condition on the uncertainty for ρ is given by $x_0(u_6 = 0) = -5$. This can be understood from Figs. 8.2a and 8.2c. They show indeed that for both negative and positive values of μ the worst case for the reliability (minimum area below the curve) is always given by $x_0 = -5$. The pictures show the particular cases $\mu = -5$ and $\mu = 5$ but the curves change monotonically as it can be understood in Fig. 8.1c. The maximum area in the worst scenario, then, always corresponds to the minimum μ.

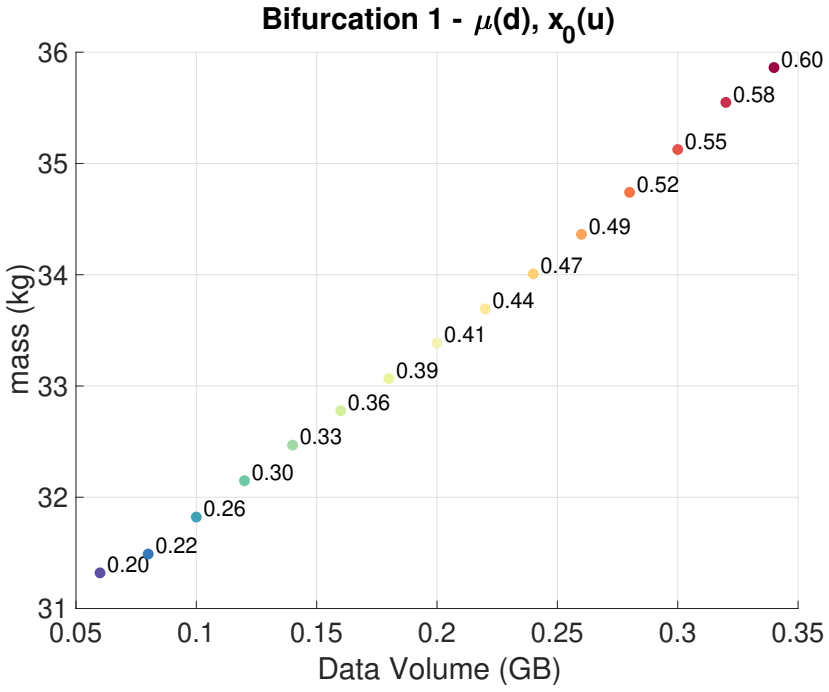
Due to the tension between performance and functionality, constraining the satellite to have an increasing g with different thresholds ν_i in Eq. (8.95), forces the solution to progressively move from $d_5 = 0.2$ to $d_5 = 0.6$. In this way the constraint can be satisfied but on the other hand the mass increases. This trade-off is well represented in Fig. 8.18a. The results with different values for the threshold ν_i are plotted in Fig. 8.18.

Super-Critical Pitchfork Bifurcation, $\mu(u_6)$, $x_0(d_5)$

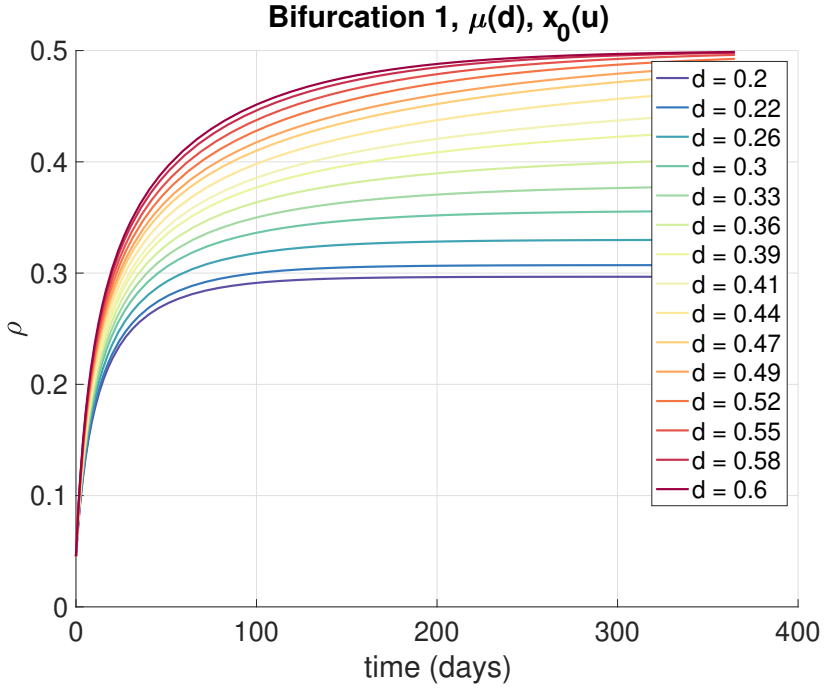
This problem is the symmetric of the previous one. The super-critical pitchfork bifurcation model in Eq. (8.19) is still used for the global reliability function ρ. However, the parameter $\mu(u_6)$ and the initial point $x_0(d_5)$ are treated as uncertain and decision variable respectively:

$$\begin{aligned} \mu(u_6) &= 0.25u_6 \\ x_0(d_5) &= 25d_5 - 10 \end{aligned} \tag{8.101}$$

In Eq. (8.101) the parameter x_0 interpolates the values $\{-5,\ 5\}$ over the design parameter $d_5 \in [0.2,\ 0.6]^T$. The parameter μ interpolates the values $\{0,\ 5\}$ over the uncertain parameter $u_6 \in [0,\ 20]^T$. In this way, the model has been set such that $x_0(d_5 = 0.2) = -5$, is the best for the mass and the worst for the reliability function. On the other side, $x_0(d_5 = 0.6) = 5$ becomes optimal for ρ but the worst for the mass.



(a) Pareto front between $mass$ and the compressed data volume DV^c_{tot}.



(b) Reliability curve ρ in the worst case for the uncertainty variable u_6.

Figure 8.18: Results of the CMOP minmax in Eq. (8.95) where the supercritical pitchfork bifurcation model in Eq. (8.19) modified as in Eq. (8.100) is used for the system dynamics. Its normalisation based on Section 8.4.2 defines the global reliability function ρ. We consider: $\mu \propto d_5$ and $x_0 \propto u_6$.

With regard to the uncertainty for $\mu(u_6)$: for $x_0 < 0$ the worst condition is $\mu = 5$ while for $x_0 > 0$ it is $\mu = 0$.

Following the Eq. (8.19), the dynamics x becomes:

$$\begin{aligned} \dot{x} &= \mu(u_6)x - x^3 \\ x_0 &= x(d_5) \end{aligned} \tag{8.102}$$

The global reliability $\rho(\mathbf{d}, \mathbf{u}, t)$ is calculated from x through the normalisation process explained in Section 8.4.2. The function $\rho(\mathbf{d}, \mathbf{u}, t)$ is used to define the functionality $g = f_V$ in Eq. (8.97) which is finally the constraint in Eq. (8.95).

The Pareto front with the corresponding values of d_5 and the reliability curves ρ are presented in Fig. 8.19.

Sub-Critical Pitchfork Bifurcation, $\mu(d_5)$, $x_0(u_6)$

In this problem, the sub-critical pitchfork bifurcation model in Eq. (8.20) is used for the global reliability function ρ. In particular, the parameter $\mu(d_5)$ is treated as design and the initial point $x_0(\mu, u_6)$ is function of both μ and u_6:

$$\begin{aligned} \mu(d_5) &= -25d_5 + 10 \\ x_0(\mu, u_6) &= -\frac{|\mu|}{20}u_6 \end{aligned} \tag{8.103}$$

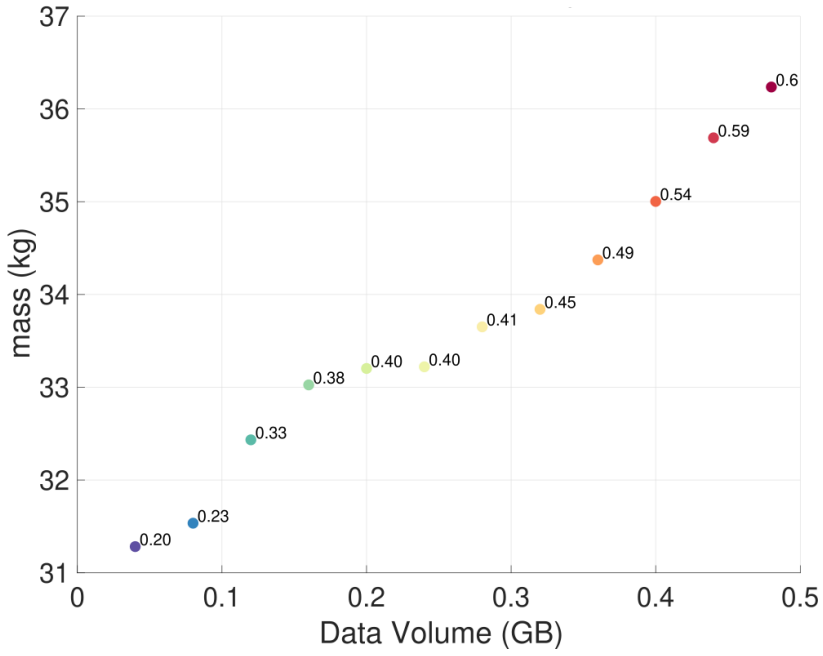
In Eq. (8.103) the parameter μ interpolates the values $\{5, -5\}$ over the parameter $d_5 \in [.2 \;\; .6]^T$, while the initial state x_0 interpolates the values $\{0, -|\mu|\}$ over the parameter $u_6 \in [0 \;\; 20]^T$. As d_5 leave the optimal value for the unconstrained problem $d_5 = 0.2$ and moves to $d_5 = 0.6$ worsening the mass, the reliability ρ increases linearly shifting the plot from Fig. 8.3d to Fig. 8.3b. However, as $|d_5|$ increases, also the uncertainty on the initial point grows.

Following the Eq. (8.20), the dynamics x becomes:

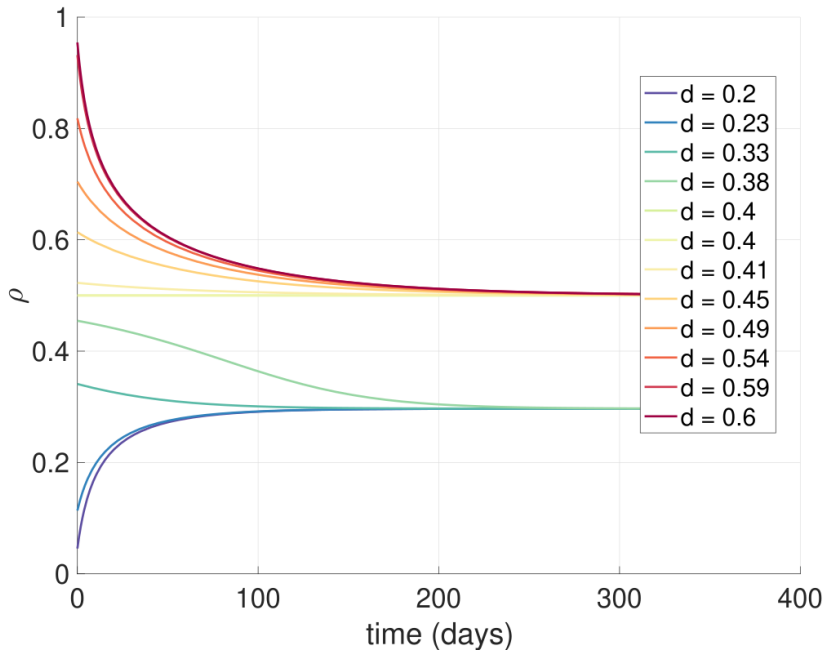
$$\begin{aligned} \dot{x} &= \mu(d_5)x + x^3 \\ x_0 &= x0(\mu, u_6) \end{aligned} \tag{8.104}$$

The global reliability $\rho(\mathbf{d}, \mathbf{u}, t)$ used in Eq. (8.19) is finally calculated from x through the normalisation process explained in Section 8.4.2. The function $\rho(\mathbf{d}, \mathbf{u}, t)$ is used to define the functionality $g = f_V$ in Eq. (8.97) which is finally the constraint in Eq. (8.95).

The results are plotted in Fig. 8.20. Up to a certain threshold $\nu < 0.25$, the optimal solution is found for values of d_5 that are close to the optimal unconstrained solution $d_5 = 0.2$ and that generate a positive μ. For bigger thresholds, instead, the optimiser is forced to choose values of d_5 far from the optimal unconstrained solution that also increases the uncertainty on the initial point x_0 with a further increase in the mass.

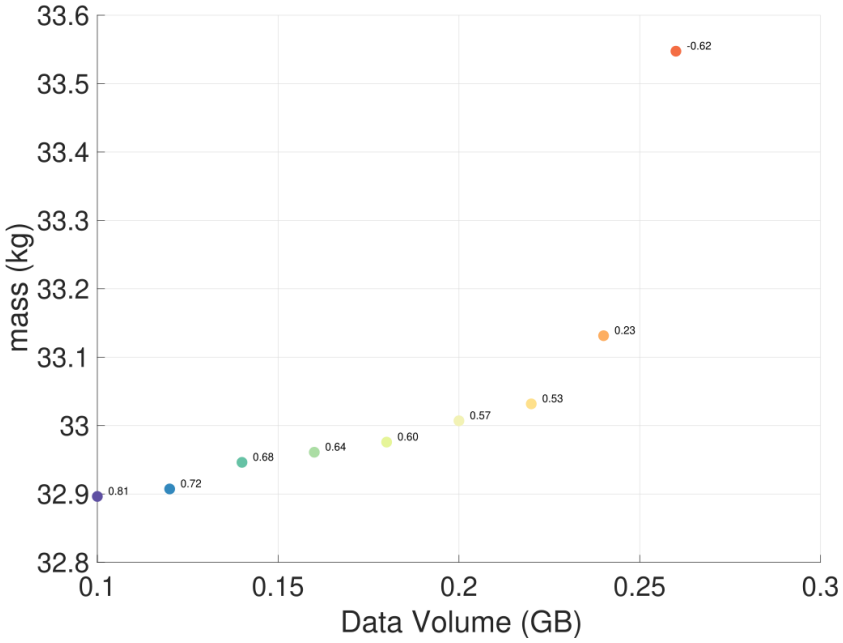


(a) Pareto front between $mass$ and the compressed data volume DV^c_{tot}

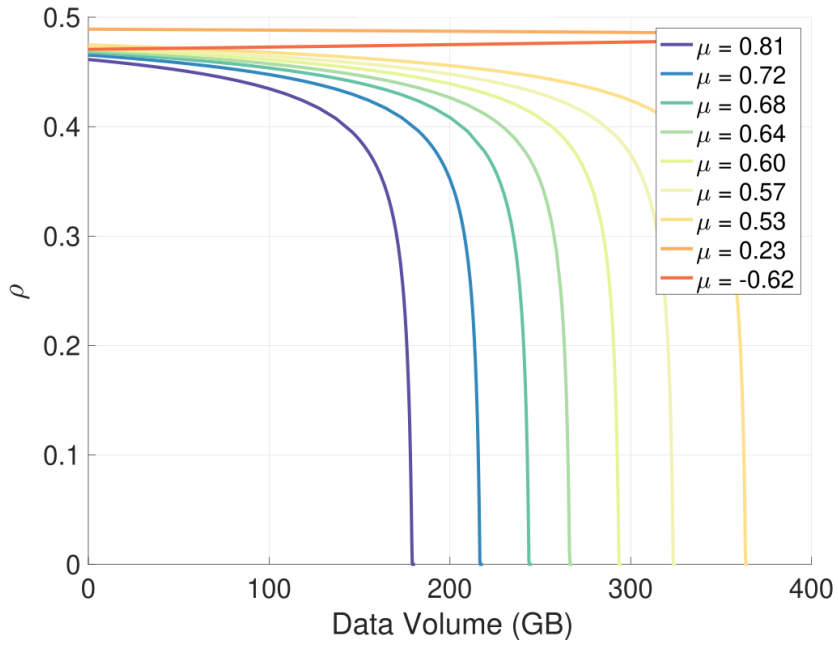


(b) Reliability curve ρ in the worst case for the uncertainty variable u_6.

Figure 8.19: Results of the CMOP minmax in Eq. (8.95) where the supercritical pitchfork bifurcation model in Eq. (8.19) modified as in Eq. (8.102) is used for the system dynamics. Its normalisation based on Section 8.4.2 defines the global reliability function ρ. We consider: $\mu \propto u_6$ and $x_0 \propto d_5$.



(a) Pareto front between $mass$ and the compressed data volume DV^c_{tot}



(b) Reliability curve ρ. Only the red curve corresponding to a negative μ is a stable solution, while the other diverge.

Figure 8.20: Results of the CMOP minmax in Eq. (8.95) where the sub-critical pitchfork bifurcation model in Eq. (8.20) modified in Eq. (8.104) is used for the system dynamics. Its normalisation based on Section 8.4.2 defines the global reliability function ρ. We consider: $\mu \propto d_5$ and $x_0 \propto u_6$.

Shock and Recovery with the Hysteresis Bifurcation, $\mu(d,t)$, $x_0(u)$

The non-autonomous bifurcation model with hysteresis in Eq. (8.21) is used here to model the global reliability function ρ. We consider the parameter $\mu(t, d_5)$ to be dependent on time and on the design variable d_5 and to follow the Eq. (8.32). The initial condition $x_0(u_6)$ is considered instead to be uncertain:

$$x_0(u_6) = 0.5u_6 - 5 \tag{8.105}$$

Following Eq. (8.21) then the dynamics x becomes:

$$\begin{aligned} \dot{x} &= \mu(d_5, t) + x - \tfrac{1}{3}x^3 \\ x_0 &= x_0(u_6) \end{aligned} \tag{8.106}$$

The global reliability $\rho(\mathbf{d}, \mathbf{u}, t)$ is calculated from x through the normalisation process explained in Section 8.4.2. The function $\rho(\mathbf{d}, \mathbf{u}, t)$ is used to define the functionality $g = f_V$ in Eq. (8.97) which is finally the constraint in Eq. (8.95).

Differently from the previous sub-problems, we propose here a single objective version of Eq. (8.95). We compare however the calculated resilient solution with the ones one would obtain without the EBORe approach. Indeed, we make a comparison between the solution of Eq. (8.95) with a single threshold $\nu = 0.2$ (*optimal-resilient*) and the optimal solution calculated with nominal values for the uncertain variables (*optimal-nominal*):

$$\begin{cases} d^* = \arg\min_{\mathbf{d}\in D} Mass(\mathbf{d}, \mathbf{u}_{nom}) \\ s.t.\ \ DV^c_{tot}(\mathbf{d}, \mathbf{u}_{nom}, t) \geq 0.2 \end{cases} \tag{8.107}$$

Eq. (8.107) is calculated considering the nominal values $\mathbf{u}_{nom}$ for the uncertain parameters equal to the mean between their lower and upper bounds. We can however explore the uncertain space to understand better the worst possible scenarios in the uncertain space. Fixing d^* we then calculate the worst condition in terms of performance f:

$$\max_{\mathbf{u}\in U} Mass(\mathbf{d}^*, \mathbf{u}) \tag{8.108}$$

and the worst condition in terms of functionality g:

$$\min_{\mathbf{u}\in U} DV^c_{tot}(\mathbf{d}^*, \mathbf{u}). \tag{8.109}$$

These two solutions are plotted respectively in green and red in Fig. 8.21. We see that the *optimal-nominal* solution has an associated risk of not satisfying the requirements in the data volume and also causing an increase in the mass of the final satellite during the design process.

We propose, finally, in orange in Fig. 8.21, the resilient solution from Eq. (8.95). It gives the minimum mass of the satellite in the worst condition while satisfying the constraint over all the possible uncertain scenarios. Looking at Fig. 8.21, the orange mass is considerably bigger than the blue one and even bigger than the green one. However, the orange design assures to satisfy the constraint, while the blue solution can lead to a drastic reduction of the data volume produced by the satellite. Furthermore,

looking at Fig. 8.21b, the resilient solution can absorb the shock in the worst scenario and recover after that, while the blue design brings to the red curve in the worst condition and represents a total failure without the possibility to recover after the disruption.

8.7 Complexity Analysis

This section gives a high-level comparison and complexity analysis of the two methods proposed for modelling resilience in the context of EBORe: HCTMC in Section 8.3 and Catastrophe Theory in Section 8.4. Both approaches model the possible transition of the system between different states, where this trajectory depends on the selected design configuration $\mathbf{d}$, the uncertain variables $\mathbf{u}$ and time t. The two modelling frameworks are connected, indeed the Catastrophe Theory model is smooth and continuous while HCTMC can be considered as a possible discretisation within the state space.

The computational complexity of the Catastrophe Theory approach includes the solution of the ODE in Eq. (8.9) (but in general of a system of ODEs in Eq. (8.13)), the generation of ρ by normalisation of x as in Section 8.4.2, the calculation of the QoI and finally the integration as in Eq. (8.16). With regards to HCTMC instead, the computational cost follows the expression in Eq. (8.8).

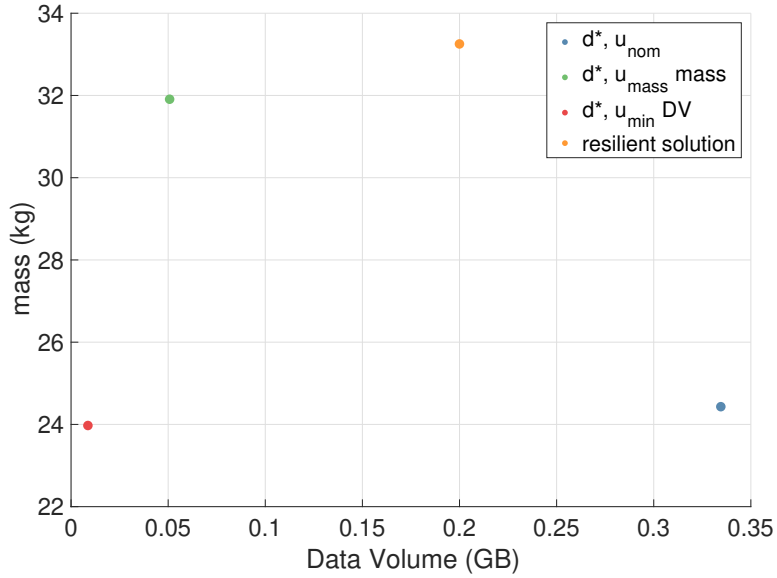
8.8 Conclusion

The chapter presented a new definition of System Resilience and Resilience Engineering. Our resilience approach combines elements of robustness and global reliability. In particular, the dynamic part of resilience is incorporated into the global reliability model. It then presented a mathematical formulation including resilience into the system optimisation under epistemic uncertainty. The method is called EBORe. The chapter then proposed two different approaches for the modelling of system resilience: the first is a discrete-state stochastic process based on HCTMC while the second is a continuous-state dynamic system model based on Catastrophe Theory.

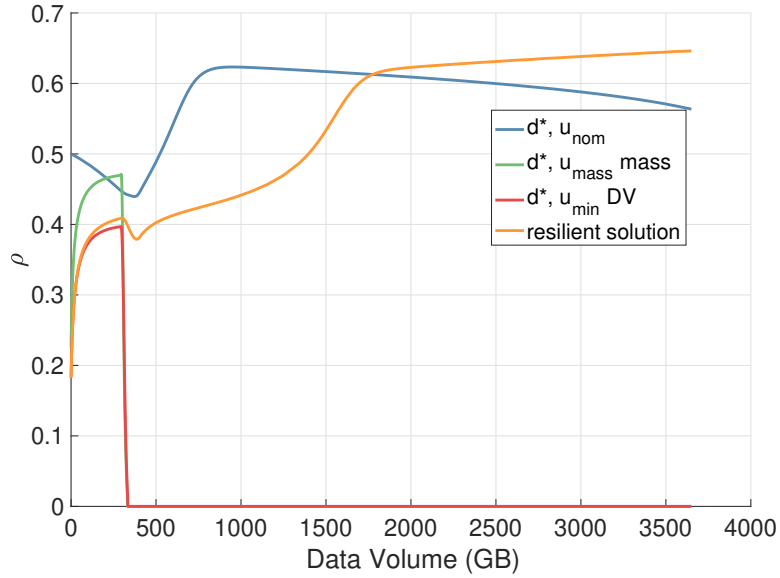
It has been shown how system resilience emerges from the combination of robustness and global system reliability.

The method has been finally applied to the design optimisation of space systems. It has been tested with a range of different sub-problems and with both resilience models. The solutions demonstrate that a design configuration can be found that is feasible, performance-optimal and resilient for all the possible realisations of the uncertain variables. This design configuration, furthermore, minimises the worst value of the objective function over the uncertain variables. Compared to a solution that uses standard safety margins, the resilient solution is better both in terms of resilience and performance. Furthermore, compared to an optimised solution that does not account for uncertainty, the resilient solution improves the number of transitions to a fully functional state.

It was also shown that the computational cost is affordable provided that subsystem performance and reliability metrics can be evaluated in a short time on a standard



(a) Bi-objective comparison between resilient and nominal solutions. The resilient solution is represented by the orange point. The nominal solution is instead represented by three instances: blue, green and red dots. The blue dot corresponds to the solution of Eq. (8.107) where d^* is the optimal nominal design and u_{nom} is the nominal uncertain vector. The green corresponds to the solution of Eq. (8.108) with $u_{max}mass$ the worst uncertain condition for the $mass$. The red point corresponds to the solution of Eq. (8.109) with $u_{min}DV$ the worst condition for the data volume DV.



(b) Global reliability curve ρ comparison between resilient and nominal solutions.

Figure 8.21: Comparison of the resilient solution and the nominal one. The former is the solution of the CSOP min-max in Eq. (8.95) where the hysteresis bifurcation model in Eq. (8.21) modified in Eq. (8.106) is used for the system dynamics. Its normalisation based on Section 8.4.2 defines the global reliability function ρ. We consider: $\mu(t, d_5)$ and $x_0(u_6)$. The nominal solution is obtained from Eqs. (8.107) to (8.109).

desktop.

9

Multi-Layer Network Model for Design Process Optimisation Under Epistemic Uncertainty

The content of this chapter was published in:

- **G. Filippi** and M. Vasile, A multi-layer evidence network model for the design process of space systems under epistemic uncertainty, in *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences (A. Gaspar-Cunha, J. Periaux, K. C. Giannakoglou, N. R. Gauger, D. Quagliarella, and D. Greiner, eds.)*, (Cham), pp. 227243, Springer International Publishing, 2021 *https://doi.org/10.1007/978-3-030-57422-2_15* [56].
- **G. Filippi** and M. Vasile, "A multi layer evidence network model for the design process of Space Systems under epistemic uncertainty", EUROGEN, 2019 [66].

This chapter introduces a new approach to the design process optimisation of complex systems affected by epistemic uncertainty. In particular, a multi-layer network called ML-ENM, is proposed to model the whole design process and the transition between adjacent design phases. Each layer represents a design phase where each node is a subsystem and each link is a sharing of information. The network is used to quantify and propagate uncertainty through the different layers (design phases). At each phase, from phase A to phase F, the fidelity of the mathematical model describing each subsystem and component is increased. Thus, it can be considered a multi-fidelity approach for the design of a complex system affected by epistemic uncertainty. The framework of DST is used to model epistemic uncertainty. We are here particularly interested in space systems. They are complex systems involving multiple interconnected components and disciplines with complex couplings: payload, structure, thermal analysis, attitude, control, etc. A system-level optimal solution cannot be found by optimising the single subsystems independently. Furthermore, the design and optimisation of

space systems have to account for epistemic uncertainty, in particular in the early design phase. In fact, knowledge about systems and requirements is only acquired incrementally, but substantial commitments are made upfront, essentially in the unknown. Even if the research field is demonstrated to be very active in proposing new and promising methodologies for the DUU of CEdS, the space industry, on the other side, has a conservative approach that is based on traditional methods. In fact, the most common and well-established approach to handling uncertainty in space systems engineering is to use safety margins and redundancies [54]. These traditional methods, however, present two critical problems that affect the result of the design process. There is a lack of an appropriate quantification of uncertainty that brings to an overestimation or an underestimation of the effect of uncertainty (increase in costs and development time or occurrence of undesirable events). There is also a lack of a holistic view of the system's performance and evolution.

In this chapter, then, we propose a methodological advancement to solve those two problems with specific applications to the design of space systems. The novelty is given by a mathematical model, in the form of a multi-layer graph, that simulates the evolution in time of the space system during the design process and is able to quantify and propagate epistemic uncertainty through the different design phases. In particular, this chapter proposes a method to propagate uncertainty through the ML-ENM from the last design phase to the first one. Then the system is optimised for robustness with a min-max algorithm [55, 67]. Evidence Theory is applied to quantify uncertainty on the optimal solution [58, 60, 61]. It is finally shown that the optimal solution at phase A is robust against the uncertainty in the next design phases. The model, ML-ENM is a generalisation of the ENM presented in Chapter 6. The ML-ENM allows for a rigorous and fast propagation of epistemic uncertainty and gives a holistic view of the whole design process. Each layer represents a different phase in the design process, each node represents a subsystem or a component at a particular level of granularity and each link is a sharing of information. The proposed contribution consists of a holistic modelling and optimisation approach for the whole design process of CEdgS which includes both CEdS and CEgS. The method takes into consideration epistemic uncertainty and decisions. Different algorithmic approaches developed in the previous chapters are here combined. The method extends the benefits presented in Chapter 7 for ENM in a multi-layer environment.

The reader finds the necessary background in Chapter 2 for a definition and description of MBSE and in particular for an explanation of the different design phases in which the design process of a CEdS is decomposed and for a description of the traditional methodology implemented by space industries, that is the margin approach. Further important information can be recovered in the previous chapters of this thesis: Chapter 5 for the bi-level optimisation approach developed and here implemented, Chapter 6 for the single-layer ENM used for UQ with DST.

The ML-ENM is proposed to model the system and the whole design process with the transition between adjacent phases and the uncertainty involved in the models. The framework of DST is used to model epistemic uncertainty. A test case is formulated and the results are presented.

Section 9.1 presents the ML-ENM and explains how it generalises ENM in Chap-

ter 6. Section 9.2 presents the optimisation method. Section 9.3 introduces to a test-case. Section 9.4 shows the results. Finally, Section 9.5 concludes the chapter.

9.1 Multi-Layer Evidence Network Model

This section introduces the concept of ML-ENM that can be used to quantify and propagate epistemic uncertainty through the complex system and the different phases of the whole design process. ML-ENM is a framework for a decomposition procedure that evaluates Belief and Plausibility curves with a computational cost that is polynomial and not exponential with the problem dimension.

ML-ENM is a multi-layer network where each layer represents a different phase in the design process, each node represents a subsystem or a component at a specific level of granularity and each link is a sharing of information. As the design process proceeds from pre-phase A to phase D, an increasing level of detail is needed in the analysis, the focus is shifted from the subsystem level to the component level, more precise mathematical models are implemented and the number of nodes increases. On the other side, studying how real projects evolve, there is a high level of confidence that between a phase and the following one, unforeseen circumstances require a modification of the design requirements and goals. Furthermore, different players collaborate on the project and, usually, good communication between them is not an easy task. Based on the results of the single design phase and on the uncertainty of the whole process evolution, the designers make decisions that bring them to the next phase and that will drive the design process.

Looking at Section 9.1, for example, during phase A three subsystems are considered and optimised. During phase B the number of considered components is increased. The point is that the number of sub-divisions and the types of the components in phase B depend on the designers' choices and each decision brings the design process to a different final solution. Also, the number of possible final configurations increases exponentially with the number of layers and possible choices that can be selected between each couple of layers.

More formally, a ML-ENM with N_L layers is a pair (G, C) where G $= \{G_\alpha; \alpha \in \{1, .., N_L\}\}$ is a family of directed and weighted graphs $G_\alpha = (X_\alpha, E_\alpha)$ and $C = \{E_{\alpha\beta} \subset X_\alpha \times X_\beta; \alpha, \beta \in \{1, 2, ..., N_L\}, \beta = \alpha + 1\}$ is the set of interconnections between nodes of different layers. The *intralayer* links in E_α represent the sharing of information between subsystems and components of the space system (complex system). The *interlayer* links in $E_{\alpha\beta}$ model the decision process tree between different design phases.

Design $\mathbf{d}$ and uncertain $\mathbf{u}$ vectors are decomposed in two components: $\mathbf{d} = [\mathbf{d}^d, \mathbf{d}^s]^T$ and $\mathbf{u} = [\mathbf{u}^d, \mathbf{u}^s]^T$ where the former ($[\mathbf{d}^d, \mathbf{u}^d]^T$) are related to the *interlayer decision* process, between a layer and the next one, and the latter ($[\mathbf{d}^s, \mathbf{u}^s]^T$) describe the *intralayer* physical model of the space *system* at a particular level of resolution.

At each layer $\alpha \in \{\text{pre-}A, B, ..., F\}$ of the ML-ENM, the performance index can be

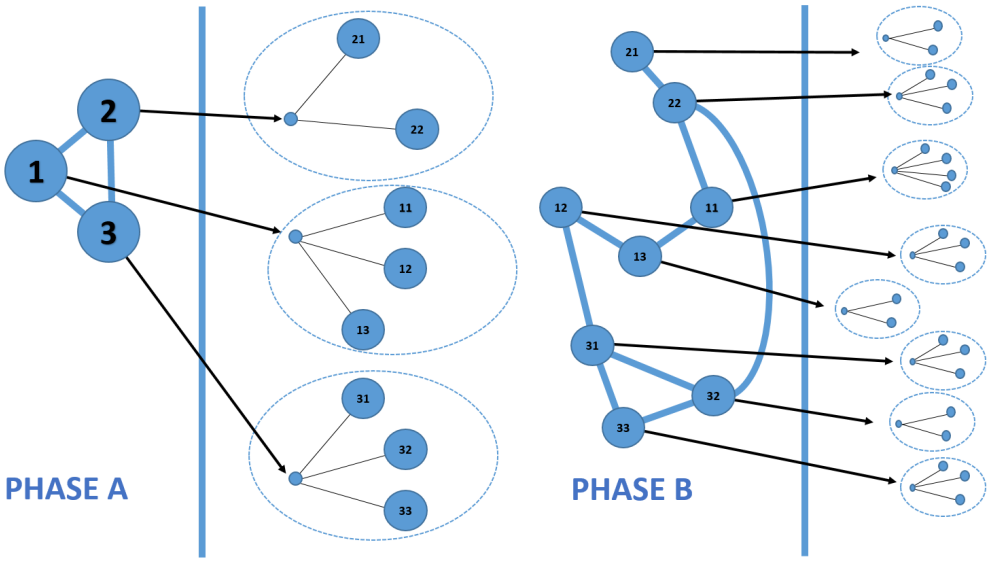


Figure 9.1: Evolution of the ENM between phase A and B: each node in phase A is decomposed into two or more nodes in phase B. The number of nodes and the mathematical model associated with them depend on the designers' choices. The process is then repeated for the next phases.

defined as:

$$f^{\alpha}(\mathbf{d},\mathbf{u}) = \sum_{i=1}^{N} g_i^{\alpha}(\mathbf{d}_i^{\alpha s},\mathbf{u}_i^{\alpha s},\boldsymbol{\varphi}_i^{\alpha s}(\mathbf{d}_i^{\alpha s},\mathbf{u}_i^{\alpha s},\mathbf{d}_{ij}^{\alpha s},\mathbf{u}_{ij}^{\alpha s})), \tag{9.1}$$

In Eq. (9.1) N is the number of nodes of the network in layer α and $\boldsymbol{\varphi}_i^{\alpha}(\mathbf{d}^{\alpha},\mathbf{u}_i^{\alpha},\mathbf{d}_{ij}^{\alpha s},\mathbf{u}_{ij}^{\alpha})$ is the vector of scalar exchange functions $\varphi_{ij}^{\alpha}(\mathbf{d}^{\alpha},\mathbf{u}_i^{\alpha},\mathbf{d}_{ij}^{\alpha s},\mathbf{u}_{ij}^{\alpha})$ that represent the input/output of the nodes, with $j \in J_i^{\alpha}$, and J_i^{α} the set of indexes of nodes connected to the i-th node of that layer. Eq. (9.1) decomposes the uncertain components $\mathbf{u}^s$ in two categories: the uncoupled components $\mathbf{u}_i^{\alpha s}$ that affect only subsystem i, and the coupled variables $\mathbf{u}_{ij}^{\alpha s}$ shared among subsystem i and one or more subsystems j.

9.2 Optimisation Approach

For the defined ML-ENM, the *Body of Evidence* presented in Sec. 3.2.2 can be populated at the last phase of the ML-ENM, here phase C, by a process of knowledge elicitation. For the proposed application, available data from previous publications has been used [193].

In this example there are only two possible paths that the design process can explore from phase A to phase C. They correspond to the choice between node 7 (*Magnetorquers*) and 8 (*Thruster*) at phase B. The choice brings, respectively, to node 19 and 20 at layer C.

For each chosen path the uncertainty structure at phase C is propagated back to

phase A exploiting the inter-layer dependencies A $\rightarrow$ B and B $\rightarrow$ C. A minimisation and a maximisation have been run to reconstruct the lower and upper bounds of each uncertain parameter at layer α that incorporate two or more parameters of layer $\alpha+1$. In this manner, the reconstructed *Body of Evidence* at phase A incorporates the uncertainty that affects the more complex and detailed models at phase B and C.

Then, the system is optimised for robustness at phase A. In particular, the min-max algorithm is used to evaluate the optimal design vector $\mathbf{d}_A^*$.

For the evaluated optimal design solution $\mathbf{d}_A^*$, the decomposition approach based on the ENM has been applied to the ML-ENM in order to propagate uncertainty through the spacecraft model and reconstruct a good approximation of the belief curve with a fraction of the computational cost required for the exact one (Fig. 9.12).

The effect of uncertainty at phases B and C is finally analysed in correspondence with the robust design solution $\mathbf{d}_A^*$.

9.3 Test Case

The ML-ENM has been here applied to the design for robustness of a spacecraft through the phases A, B and C (pre-phase A is considered in the figures for clarity). Each node of the ML-ENM is associated to a mathematical function modelling a subsystem or a component. Their list and the classification between the different phases is presented in Tab. 9.1. The quantity of interest is the overall mass of the satellite and it is given by the sum of the masses of all the subsystems (phase A) or components (phase B and C). The network can be visualised in Figs. 9.2, 9.3, 9.4 and 9.5: the nodes correspond to the models of the system (node 1 at pre-phase A), sub-systems (nodes 2-6 at phase A) and components (nodes 7-18 at phase B and nodes 19-30 at phase C). The links, instead, correspond to their intra-layer and inter-layer connections. In particular coloured arrows define inter-layer (hierarchical) dependencies while grey lines indicate intra-layer dependencies. Red lines show the dependence of nodes at layer A from the node at layer pre-A (pre-$A \rightarrow A$), yellow lines show the dependence of nodes at layer B from nodes at layer A ($A \rightarrow B$) and purple lines of nodes at layer C from layer B ($B \rightarrow C$). Each node in a generic layer, in fact, can be decomposed in two or more nodes in the next layer. Furthermore, the number of parameters and the complexity increase through the process as Tab. 9.2 shows. Gray lines instead represent couplings between nodes in the same layer α through the linking functions φ_i^α as in Eq. (9.1).

9.4 Results

Considering the first path in the ML-ENM (node 7 at phase B and node 19 at phase C), the worst case optimal solution $(\mathbf{d_1^*}, \mathbf{u_1^*})$ gives a mass of 166.43 kg. Figs. 9.6 to 9.8 show the effect of uncertainty at phases A, B and C for the fixed $\mathbf{d_1^*}$. The second path (node 8 at phase B and node 20 at phase C) brings to the robust solution $(\mathbf{d_2^*}, \mathbf{u_2^*})$ with a corresponding mass of 230.12 kg. Figs. 9.9 to 9.11 show the effect of uncertainty at phases A, B and C for the fixed $\mathbf{d_2^*}$. In particular, Figs. 9.6 and 9.9 concern the system level (the whole mass of the satellite), Figs. 9.7 and 9.10 the sub-systems level

Table 9.1: ML-ENM nodes

Node	Pre-Phase A
1	Spacecraft
	Phase A
2	Attitude and Orbit Control (AOCS)
3	Payload
4	Power
5	Thermal
6	Telemetry and Telecommand (TTC)
	Phase B
7	Magnetorquers
8	Thrusters
9	Reaction Wheels
10	Payload
11	Batteries
12	Harness
13	Power Conditioning and Distribution Unit (pcdu)
14	Solar Array
15	Thermal
16	Antenna
17	Radio Frequency Distribution Network (rfdn)
18	Transponder
	Phase C
19	Magnetorquers
20	Thrusters
21	Reaction Wheels
22	Payload
23	Batteries
24	Harness
25	Power Conditioning and Distribution Unit (pcdu)
26	Solar Array
27	Thermal
28	Antenna
29	Radio Frequency Distribution Network (rfdn)
30	Transponder

Table 9.2: Model dimension

	Path 1		Path 2	
Phase	dim_d	dim_u	dim_d	dim_u
A	6	16	6	17
B	13	35	14	34
C	21	43	21	47

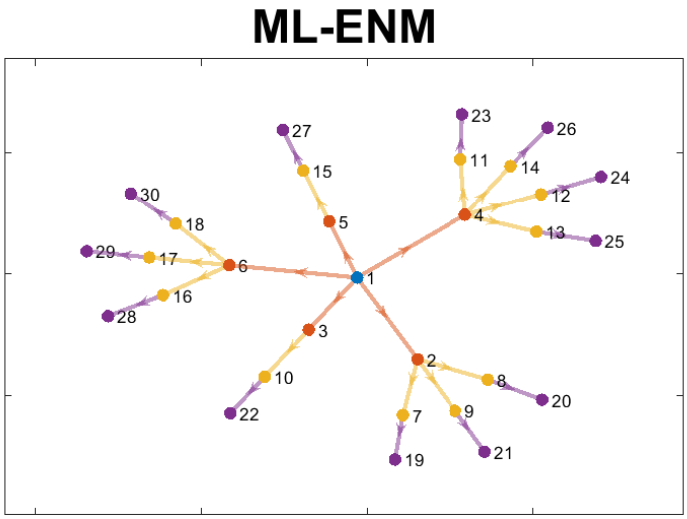


Figure 9.2: 2D representation of the design process as a decision tree. The phases (A, B and C) are indicated with different colours. Subsystem's and component's models are represented as nodes.

and Figs. 9.8 and 9.11 the components level. The box-plots have been evaluated with a Monte Carlo simulation over the uncertain space with 10^6 function evaluation. For each box a maximisation and a minimisation have been run in order to be sure that the boxes include all the possible values of the mass for the given uncertainty structure. These figures show that the spacecraft model at phase A with the back-propagation of uncertainty, incorporates for the chosen path all the uncertainty in phases B and C. The worst case optimal solution $\mathbf{d}^*$ at phase A, then, results to be robust through the design process.

For $\mathbf{d}_1^*$, finally, the decomposition approach has been applied to the ML-ENM and Fig. 9.12 presents the reconstructed belief curve. The decomposition method allows to fast and good evaluation of the belief. In this problem, in fact, the exact evaluation of the belief curves require 65536 maximisations (one for each focal element). Instead, the curve in Fig. 9.12 has been evaluated with 234 maximisations (0.36 %).

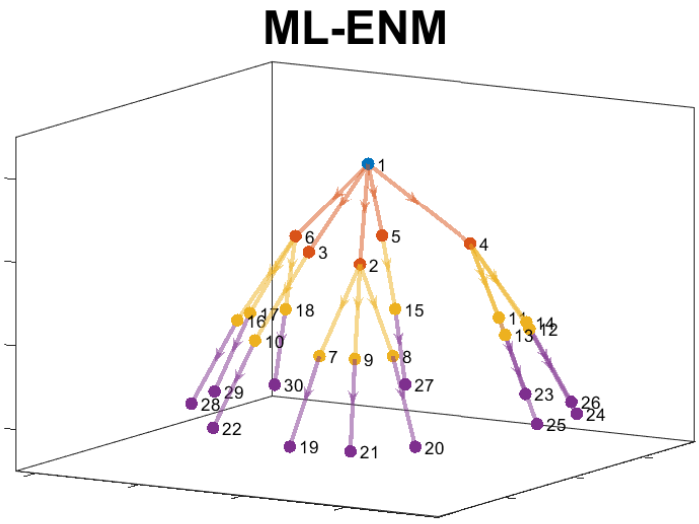


Figure 9.3: 3D representation of the design process as a tree.

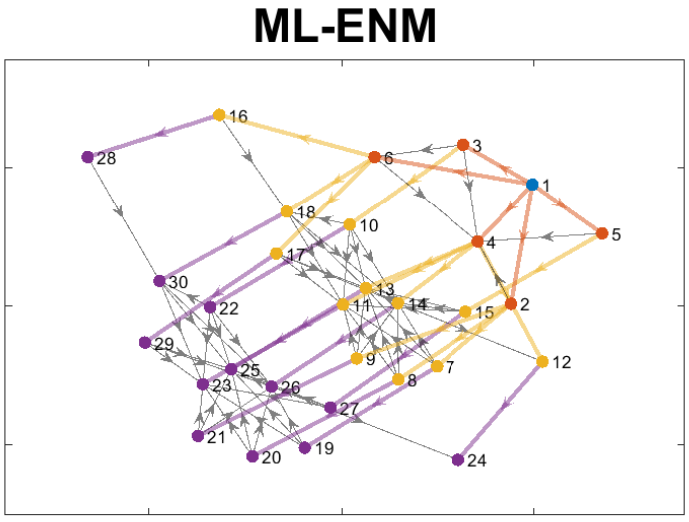


Figure 9.4: Representation of the design process as a graph. Coloured arrows define inter-layer dependencies while grey lines indicate intra-layer dependencies within the same design phase.

9.5 Conclusion

This chapter proposes a new approach for the design process of a space system affected by epistemic uncertainty. The main novelty is given by the use of the ML-ENM to quantify and propagate uncertainty between different phases of the design process.

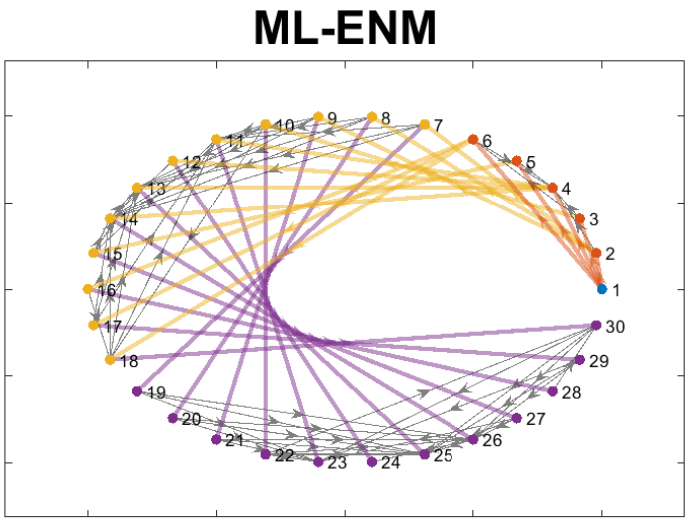


Figure 9.5: Circular representation of the ML-ENM with both inter-layer and intra-layer dependencies.

ML-ENM is a multi-layer network representation of the complex system where each layer takes into account the couplings between subsystems (or components) at a particular design phase. The evolution of the design process is then modelled by the sequence of layers.

It is here presented a method for the definition of uncertainty at the first phase (phase A) of the process such that the optimal solution at that phase is robust against the uncertainty in the following phases.

The method is applied to the design of a space system. The model is optimised for robustness and finally a decomposition methodology is applied to the network in order to reduce the computational cost of the epistemic uncertainty propagation and the belief reconstruction with the use of Evidence Theory.

It has be shown that the optimal design solution at phase A defined in such a way, is robust against the propagation of uncertainty through the design process.

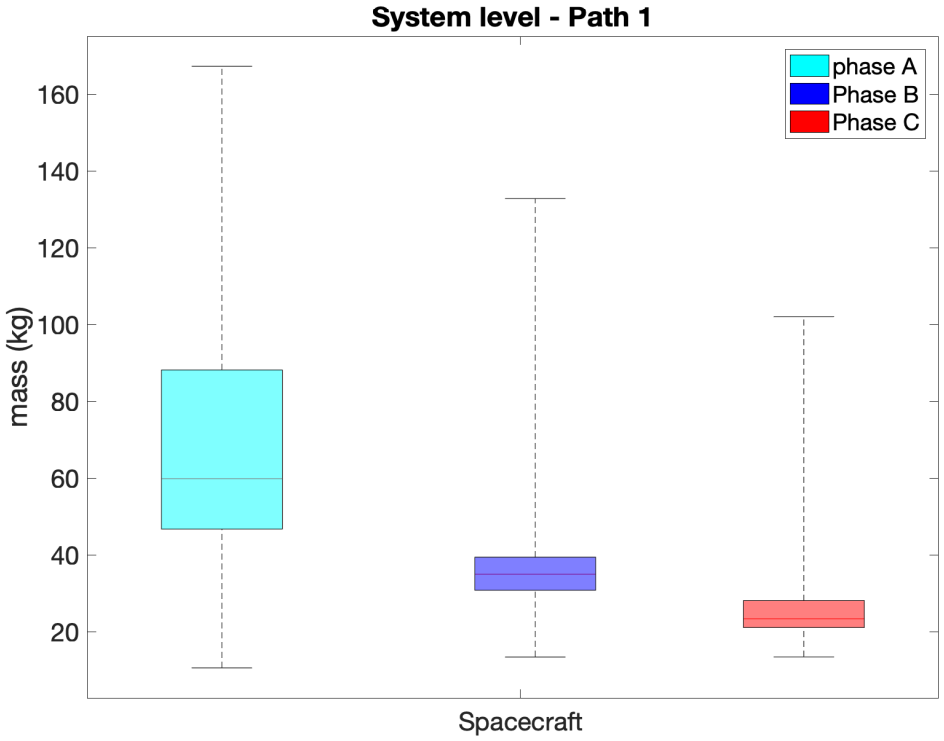


Figure 9.6: Effect of uncertainty at the system's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution.

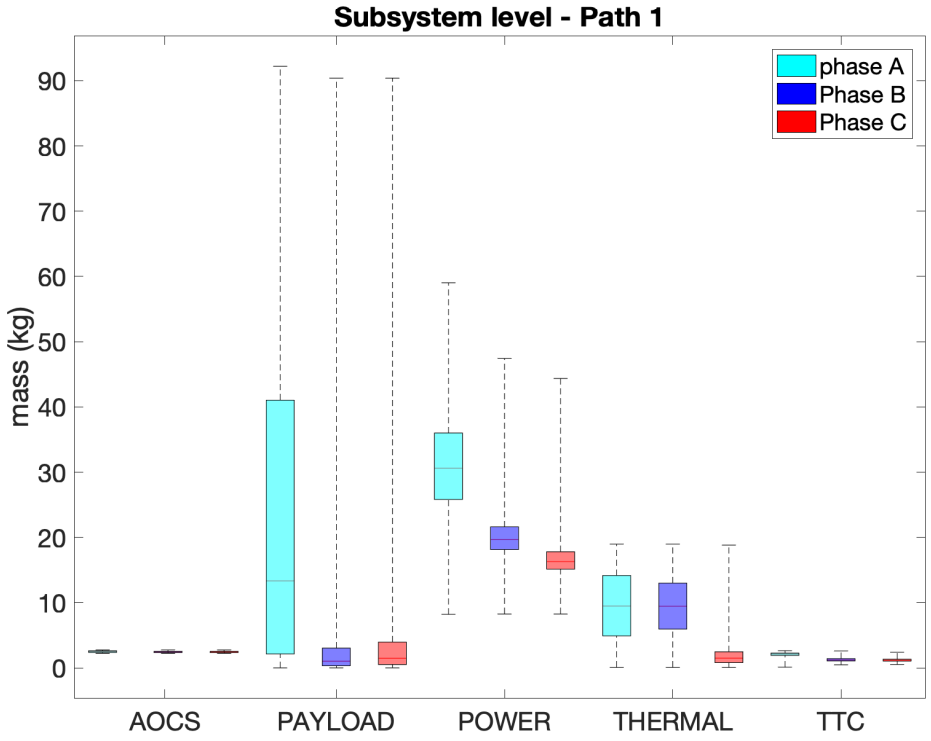


Figure 9.7: Effect of uncertainty at the sub-system's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution.

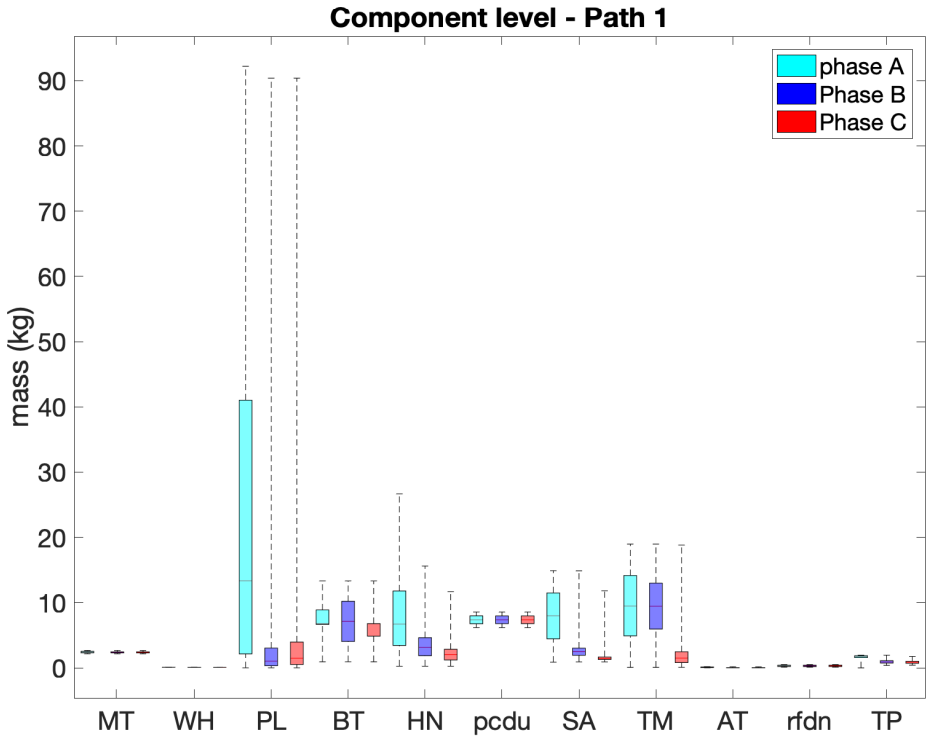


Figure 9.8: Effect of uncertainty at the component's level in phases A, B and C for the first considered path. The design vector is fixed at the optimal solution.

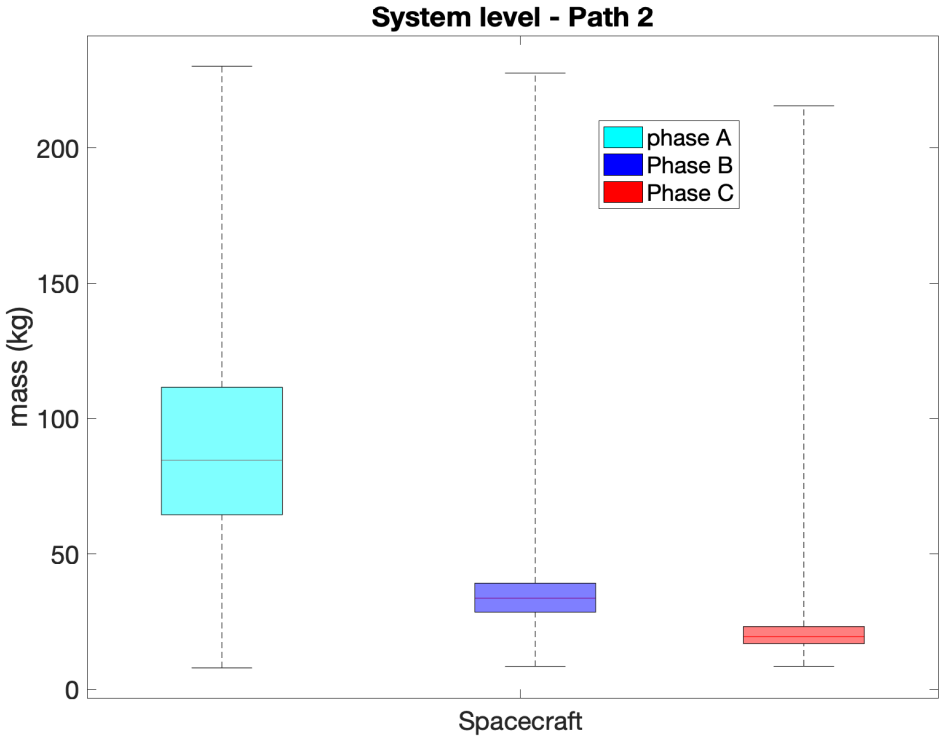


Figure 9.9: Effect of uncertainty at the system's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution.

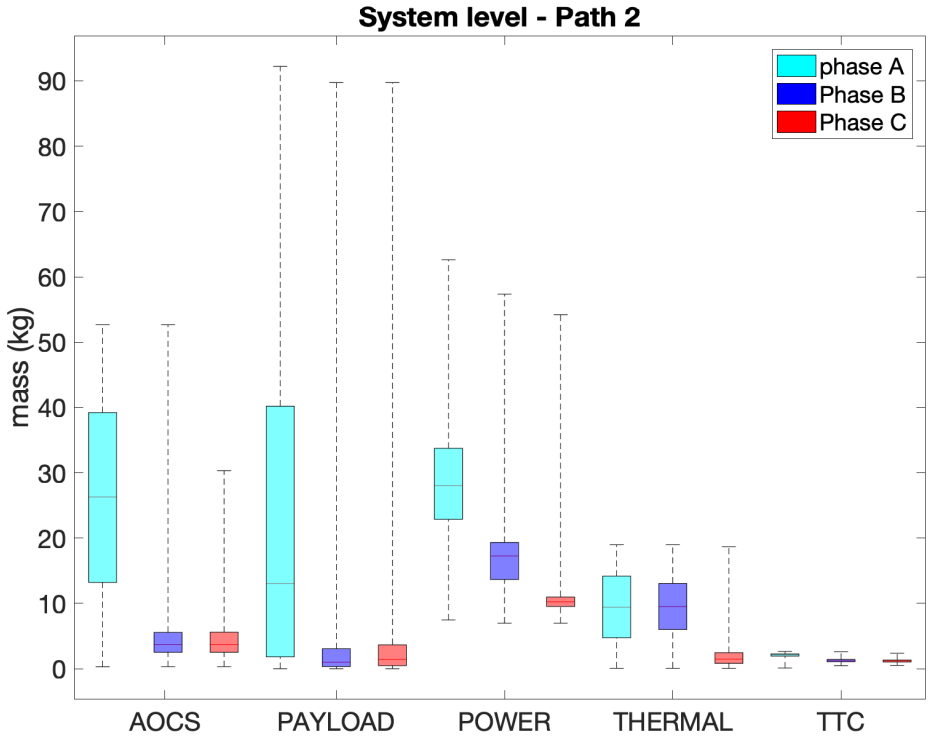


Figure 9.10: Effect of uncertainty at the sub-system's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution.

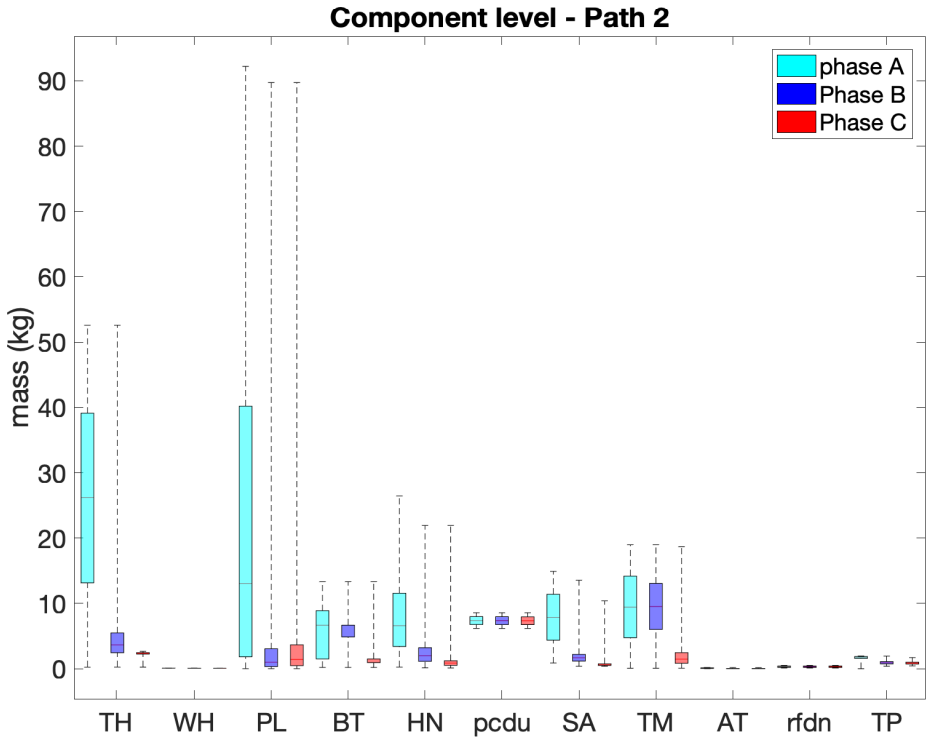


Figure 9.11: Effect of uncertainty at the component's level in phases A, B and C for the second considered path. The design vector is fixed at the optimal solution.

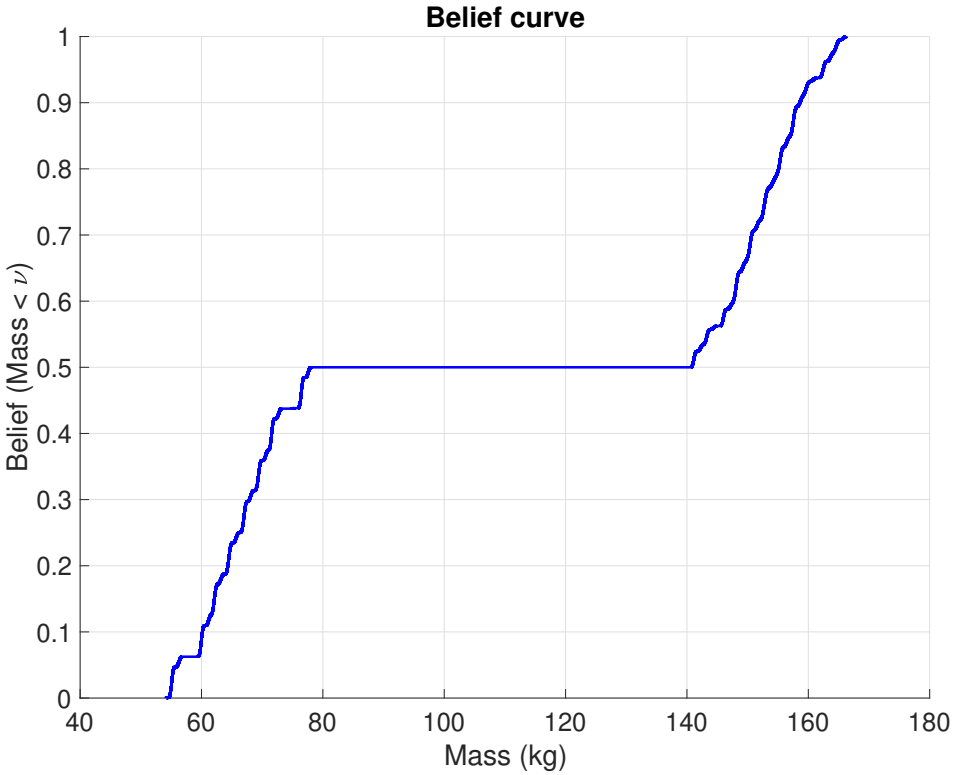


Figure 9.12: Cumulative Belief curve of the optimal worst case solution at phase A for the first considered path.

9. Multi-Layer Network Model for Design Process Optimisation Under Epistemic Uncertainty

9. Multi-Layer Network Model for Design Process Optimisation Under Epistemic Uncertainty

10

Conclusion

" Begin at the beginning, the King said, gravely,
and go on till you come to an end;
then stop. "

– Lewis Carroll, *Alice in Wonderland*

This thesis had the goal of developing a novel SDO approach for the design of CEdgS under epistemic uncertainty that can be applied within the MBSE framework. We defined the CEdgS as the combination of CEdS and CEgS being the former the physical system and the latter the design process used to produce it. As explained in Part I, we are currently in the age of complexity and we need a paradigm shift from traditional methodologies to more suitable ones. The work presented in the thesis went in this direction and its contribution was twofold. On one side we developed a set of new tools to solve specific problems in system modelling, CGO, UQ and OUU. On the other side, we presented a holistic SDO framework that couples and orchestrates the different building blocks. The reader can find the list of objectives of the thesis in the introductory Chapter 1.

With regard to the first objective in Chapter 1, our contribution on the modelling framework for CEdgSs was presented in Chapters 6 and 9.

We showed in Chapter 6 the ENM for the modelling of a generic CEdS. Then, we generalised the ENM to cope with the CEdgSs by including also the evolution of the CEdS through the different phases of the design process (the CEgS). We called this generalisation ML-ENM and we presented it in Chapter 9.

We demonstrated the capability of the framework to capture the complexity of a generic CEdgS. Complexity here was considered to derive from the combined effect of interconnections between subsystems and their dependency on uncertain variables. The proposed framework reached this objective thanks to a network structure where sub-systems were modelled as nodes of the graph and their inter-relation and inter-correlation were translated into links. Also, to both nodes and links a function was defined that depended on the uncertain vector of variables.

We found that the network structure of ENM is efficient for the application of SDO and decomposition methods for UQ as presented later in Chapter 7. Indeed the ENM splits uncertain variables in 'coupled' and 'uncoupled' vectors and represents relations between nodes through coupling functions.

Compared to traditional MDO methods, we showed that within an ENM the correlations between nodes are represented by scalar values that model in a compact way the influence of many uncertain parameters and weight the different links. We also found that the ENM permits an easier representation of sub-networks and clusters.

We demonstrated the feasibility of both ENM and ML-ENM with realistic benchmarks and test cases.

Even if real engineering problems are more complicated and complex than the ones here developed and used, we consider the objective to be achieved since it referred to the feasibility proof of the methodology.

The proposed method did not present direct limitations. Since it was open and agnostic to the specific models used for each sub-system, the limitations could come only from these models.

With regard to the second objective, the main contribution of this thesis was presented in Section 7.2. It was a new algorithmic approach for UQ to quantify epistemic uncertainty modelled with DST. It was obtained through an iterative SDO decomposition procedure coupled with the ENM in Chapter 6. A partial and side contribution is also presented in Appendix C and refers to an alternative decomposition method based on an evolutionary Binary Tree. The following comments will only refer to the first method.

We demonstrated in Eq. (7.13) that our decomposition approach in Section 7.2 coupled with the ENM reduces the computational complexity of UQ with DST from exponential to polynomial with respect to the number of interacting sub-system's components. The reduction of computational complexity is the major challenge of UQ with DST for real systems and in general for high dimensional models. As stated in Chapter 3 indeed, the computational cost arises from the total number of FE (optimisation runs) that needs to be performed.

We showed that both approaches are outer approximations, meaning that they produce conservative results.

We also showed that they are iterative procedures that produce estimated values of Belief and Plausibility which eventually converge to the exact solution with an appropriate number of iterations.

We demonstrated the feasibility of the method on a set of test cases and on a realistic problem application.

We showed the scalability behaviour of the approach by increasing the number of sub-systems (nodes in the network) and the number of variables.

The main limitation of the presented decomposition approach for UQ with DST based on the ENM was related to the restriction of its applicability to the class of M-decomposable problems with $M = 0$ as stated in Chapter 6. This class of problems is however frequent in aerospace engineering.

With regard to the third objective, we presented our contribution in Chapter 8. It included original definitions and metrics.

We presented a novel definitions of 'Resilience' and 'Resilience Engineering' and we explained how and why they combined both the concepts of 'Reliability' and 'Robustness'.

We presented two novel modelling frameworks for the Resilience metric. The first

was based on a stochastic process and in particular on HCTMC. The second was based on Catastrophe Theory.

We demonstrated that both approaches are able, in different ways, to model the system dynamics and its dependency on design and uncertain variables. We showed indeed that the HCTMC-based model considered a discrete transition between possible states while the catastrophe-based one could be seen as a generalisation of the former where the number of system's states became infinity, or alternatively, the transition was smooth.

We demonstrated that the use of Catastrophe Theory is able to model 'Black Swan' events where a drastic qualitative change in the dynamics of the system can be driven by a small change in one (or more) internal or external factors. We coupled this with epistemic uncertainty since part of the model variables was of irreducible uncertainty.

We demonstrated the feasibility of both Resilience models through realistic engineering test problems.

Concerning the limitations of the methodology, the HCTMC approach was restricted to employing only three possible states for the system: fully functioning, partially functioning, and failed. While a greater number of states could enhance the precision and fidelity of the model, it is crucial to note that this wouldn't deviate from our primary objective, which was to showcase a method for discrete state modelling of the system dynamics influenced by both design and epistemic uncertainty variables.

The HCTMC model was constructed based on the homogeneity property, meaning that the transition probability does not change in time. This condition was important in order to avoid over-complication of the metric derivation. However, it is not necessary a realistic one since it is possible for real applications to be better modelled by non-homogeneous chains.

With the HCTMC method, the global metric was evaluated through the expected value operator. However, for an entirely IP-based method, this aspect should be generalised.

Both the Resilience models were applied to a single subsystem in the ENM. We found that even in this case the result of the OUU was of interest. Such a defined approach is however limited and not entirely realistic since it is missing the study of the whole system dynamics emerging from the interaction of the dynamics of all the nodes based on the network topology. This generalisation would however introduce computational problems and require further methodological research.

Finally, the global quality of the approach was dependent on the accuracy of the models for each single sub-system or discipline. For the approach to be more realistic, future research effort needs to be put on their definition.

With regard to the fourth objective, our contribution about the optimisation methods was presented in Part II (Chapter 5). We demonstrated a new procedure for constrained OUU for CGO.

We worked on one of the possible approaches for OUU by formulating a min-max COP where the worst-case condition given by uncertainty is optimised over the design space. We stated that this approach is equivalent to the search of a Nash equilibrium in a two-players zero-sum game.

We showed that the min-max procedure is the most conservative approach in pres-

ence of uncertainty.

We demonstrated the feasibility of our min-max algorithm through synthetic test functions and a realistic engineering problem. The strategy was based on the breaking of the original AAO nested problem into a bi-level decomposition procedure. We indeed proved that the tension between the two different levels with opposing goals, one working on the design space and the other on the uncertain one brings to an iterative refinement that eventually converges the global optimum. We also found that for reasonably complex problems, the approach avoid the Red Queen Effect.

We developed such a min-max framework to be optimiser-agnostic. Indeed, we tested it with both local and global optimisation solvers.

The most important contribution about this objective, was the introduction of the constraint handling in the problem formulation. This point required an algorithmic procedure for both the overarching bi-level min-max framework and the single layer optimisation solver used within the framework. For the latter, in particular, we presented an extension of a previously developed GO solver MP-AIDEA by including a constraint handling approach. We demonstrated that the technique was able to handle constraints and satisfy them globally.

With regard to the min-max framework, we presented a constraint relaxation method that automatically updates the problem in case no feasible solution exists with the initial constraint definition.

We developed a new benchmark that can be used for the testing of the globally constrained min-max problem. The entire set of results can be found in Appendix A. With reference to the testing procedure, we proposed a generalisation of the SR metric in Section 5.4.2.

We demonstrated the feasibility of the method on a realistic engineering problem.

A further contribution refers to the computational complexity analysis of the algorithm given in Section 5.3.

As a side and partial result, we also demonstrated in Appendix B the extension of the min-max approach for the CMOP.

Concerning the method limitations, the OUU approach considered only the optimisation of the worst case scenario while other UQ metrics, such as the Belief and Plausibility were calculated as a post-processing. We demonstrated however these two methodologies (min-max and Belief quantification) independently and their coupling is left for a possible future research project.

With regard to the last objective, our contribution on the holistic framework for CEdgSs was presented in Part III. It combined the building blocks presented above to generate a SDO approach for MBSE. In particular, Chapters 7 to 9 introduced respectively to EBRO, EBORe and the optimisation method of the design process.

We showed how the 'Design for Robustness' of a CEdS can be translated in a global constraint OUU and how it can be solved by means of the developed min-max strategy. This brought to our EBRO method in Chapter 7.

We found then that the application of the developed UQ methodology allows us to understand properly the effect of uncertainty on the optimal solution when there is lack of knowledge.

Chapter 8 extends Chapter 7 by introducing the concepts of "Resilience", "Re-

silience Engineering" and "Resilience Optimisation". The chapter proved that the methodology called EBORe is able to merge UQ with CGO in the context of the CEdS dynamics to drive the design decision to an optimal solution that takes into consideration the complexity of the system under design.

We finally demonstrated that the modelling framework and the developed algorithms for UQ and OUU can be coupled to understand how uncertainty propagates through the evolution of the design process.

10.1 Future Works

The main goal of this thesis, as stated above and in the introduction, was to propose a new method and to prove its feasibility for the SDO of CEdgSs with epistemic uncertainty. Under the limitations explained above, the goal was achieved and the objectives were completed. The following of this chapter will state the main directions of future research.

Higher fidelity and more complex models of the CEdS can be developed to stress-test the methodology and use it in real engineering context. This includes two aspects. On one side, for a single design phase, we will increase the number of alternative engineering technologies and sub-systems. On the other, we will develop more fidelity levels to cope with all the phases of the design process.

Our UQ approach can be generalised to further reduce the computational complexity and to overcome the current limitations. Possible directions would include the integration of our method with MDO approaches in the literature, the use of techniques to reduce the number of FE based on their impact on the final Belief and the coupling of the two decomposition methodologies proposed in this thesis (one in Appendix). In this regard, some preliminary results have been already obtained. They were however not included in the thesis because they needed some further refinement. An other important generalisation will refer to the merging of epistemic and aleatory uncertainty and to the fusion of different, discordant and heterogeneous sources of evidence. Furthermore, since much of the inputs at least in the first preliminary phases of a design process come form experts knowledge, it would be interesting to develop and couple methods for knowledge elicitation.

The models for resilience for each of the sub-systems can be further refined. They can also be applied to all nodes of the ENM to study the emergence of the system dynamics.

With reference to the optimisation approaches, the algorithms for the solution of CMOP within the SDO approach are open to further developed. The worst-case optimisation, even if highly informative can be generalised or combined with less conservative methods. Surrogates are finally very important tools when dealing with computationally demanding simulations and they will also have an important role to play in the future development of this thesis' concepts.

Finally, it would be interesting, but certainly not within the scope of this thesis work, to build a set of benchmark problems for robust and resilience optimisation with epistemic uncertainty to allow the comparison of the methodologies available in the

literature and future ones. The test problems presented in the thesis can serve as a first step in this direction.

10. Conclusion

10. Conclusion

Appendix A

Testing Results for the Single-Objective Min-Max Problem

Table A.1: GFc-1, only fmincon, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	11000	12000
tc-1	0.73	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
tc-2	0.23	0.25	0.51	0.78	0.92	0.95	0.96	0.99	0.99	1.00	1.00	1.00
tc-3	0.01	0.12	0.43	0.56	0.85	0.98	0.99	0.99	1.00	1.00	1.00	1.00
tc-4	0.01	0.03	0.23	0.42	0.73	0.93	0.99	1.00	1.00	1.00	1.00	1.00
tc-5	0.63	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
tc-6	0.00	0.00	0.06	0.15	0.31	0.44	0.63	0.79	0.90	0.94	0.97	0.98
tc-7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
tc-8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
tc-9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
tc-10	0.16	0.22	0.30	0.36	0.44	0.51	0.53	0.55	0.56	0.58	0.60	0.65
tc-11	0.09	0.14	0.17	0.21	0.24	0.26	0.28	0.29	0.30	0.32	0.32	0.32
tc-12	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
tc-13	0.00	0.66	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table A.2: GFc-1, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.56	1.00	1.00	1.00	1.00	1.00	6.00	9.00
MWP-2	0.00	0.00	0.40	0.98	1.00	1.00	1.00	1.00	6.00	29.00
MWP-3	0.00	0.00	0.06	1.00	1.00	1.00	1.00	1.00	9.00	26.00
MWP-4	0.00	0.00	0.00	0.88	1.00	1.00	1.00	1.00	9.00	15.00
MWP-5	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	7.00	10.00
MWP-6	0.00	0.00	0.00	0.52	1.00	1.00	1.00	1.00	10.00	22.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.10	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	8.00	21.00
MWP-10	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	12.00	29.00
MWP-11	0.00	0.06	0.06	0.12	0.14	0.14	0.14	0.14	-	-
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.04	1.00	1.00	1.00	1.00	1.00	8.00	29.00
GFf-1	0.00	0.00	0.68	1.00	1.00	1.00	1.00	1.00	6.00	29.00

Table A.3: GFc-1, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.56	1.00	1.00	1.00	1.00	1.00	6.00	9.00
MWP-2	0.00	0.00	0.40	0.96	1.00	1.00	1.00	1.00	6.00	29.00
MWP-3	0.00	0.00	0.04	1.00	1.00	1.00	1.00	1.00	9.00	18.00
MWP-4	0.00	0.00	0.00	0.88	1.00	1.00	1.00	1.00	11.00	18.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	8.00	10.00
MWP-6	0.00	0.00	0.00	0.56	1.00	1.00	1.00	1.00	10.00	22.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.10	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	11.00	12.00
MWP-10	0.00	0.02	0.02	1.00	1.00	1.00	1.00	1.00	13.00	29.00
MWP-11	0.00	0.14	0.20	1.00	1.00	1.00	1.00	1.00	11.00	29.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.08	1.00	1.00	1.00	1.00	1.00	29.00	29.00
GFf-1	0.00	0.00	0.68	0.98	0.98	0.98	0.98	0.98	-	-

Table A.4: GFc-2, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.64	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-2	0.00	0.00	0.44	1.00	1.00	1.00	1.00	1.00	6.00	12.00
MWP-3	0.00	0.00	0.10	1.00	1.00	1.00	1.00	1.00	9.00	15.00
MWP-4	0.00	0.00	0.00	0.82	1.00	1.00	1.00	1.00	10.00	15.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-6	0.00	0.00	0.00	0.70	1.00	1.00	1.00	1.00	10.00	15.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.12	0.98	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	5.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	21.00
MWP-10	0.00	0.02	0.02	1.00	1.00	1.00	1.00	1.00	13.00	24.00
MWP-11	0.00	0.02	0.02	0.20	0.22	0.22	0.22	0.22	-	-
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.18	1.00	1.00	1.00	1.00	1.00	7.00	9.00
GFf-1	0.00	0.00	0.46	0.96	1.00	1.00	1.00	1.00	14.00	29.00

Table A.5: GFc-2, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.54	1.00	1.00	1.00	1.00	1.00	6.00	9.00
MWP-2	0.00	0.00	0.50	1.00	1.00	1.00	1.00	1.00	6.00	15.00
MWP-3	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	9.00	15.00
MWP-4	0.00	0.00	0.00	0.82	1.00	1.00	1.00	1.00	8.00	14.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-6	0.00	0.00	0.00	0.64	1.00	1.00	1.00	1.00	9.00	15.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.08	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	8.00	8.00
MWP-10	0.00	0.02	0.02	1.00	1.00	1.00	1.00	1.00	12.00	29.00
MWP-11	0.00	0.14	0.18	0.98	0.98	0.98	0.98	0.98	-	-
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.14	1.00	1.00	1.00	1.00	1.00	7.00	9.00
GFf-1	0.00	0.00	0.62	1.00	1.00	1.00	1.00	1.00	28.00	29.00

Table A.6: GFc-3, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	6.00	9.00
MWP-2	0.00	0.00	0.50	1.00	1.00	1.00	1.00	1.00	6.00	14.00
MWP-3	0.00	0.00	0.10	0.94	1.00	1.00	1.00	1.00	9.00	29.00
MWP-4	0.00	0.00	0.04	1.00	1.00	1.00	1.00	1.00	8.00	11.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	8.00	9.00
MWP-6	0.00	0.00	0.00	0.54	1.00	1.00	1.00	1.00	10.00	16.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.26	0.76	0.80	-	-
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	0.90	0.98	1.00	1.00	1.00	1.00	1.00	9.00	29.00
MWP-10	0.00	0.02	0.02	1.00	1.00	1.00	1.00	1.00	11.00	29.00
MWP-11	0.00	0.02	0.28	1.00	1.00	1.00	1.00	1.00	9.00	29.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.42	1.00	1.00	1.00	1.00	1.00	7.00	10.00
GFf-1	0.00	0.00	0.44	0.94	0.98	1.00	1.00	1.00	19.00	29.00

Table A.7: GFc-3, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	6.00	9.00
MWP-2	0.00	0.00	0.44	1.00	1.00	1.00	1.00	1.00	7.00	16.00
MWP-3	0.00	0.00	0.00	0.94	1.00	1.00	1.00	1.00	9.00	21.00
MWP-4	0.00	0.00	0.10	1.00	1.00	1.00	1.00	1.00	7.00	11.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-6	0.00	0.00	0.00	0.68	1.00	1.00	1.00	1.00	8.00	15.00
MWP-7	0.00	0.00	0.00	0.00	0.02	0.54	1.00	1.00	18.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	18.00
MWP-10	0.00	0.02	0.02	1.00	1.00	1.00	1.00	1.00	12.00	29.00
MWP-11	0.00	0.16	0.28	1.00	1.00	1.00	1.00	1.00	10.00	29.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.34	1.00	1.00	1.00	1.00	1.00	7.00	10.00
GFf-1	0.00	0.00	0.60	0.96	1.00	1.00	1.00	1.00	19.00	29.00

Table A.8: GFc-4, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.86	1.00	1.00	1.00	1.00	1.00	6.00	19.00
MWP-2	0.00	0.00	0.64	1.00	1.00	1.00	1.00	1.00	6.00	29.00
MWP-3	0.00	0.00	0.00	0.84	1.00	1.00	1.00	1.00	10.00	29.00
MWP-4	0.00	0.00	0.02	0.98	1.00	1.00	1.00	1.00	8.00	29.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	13.00
MWP-6	0.00	0.00	0.00	0.44	0.98	1.00	1.00	1.00	11.00	18.00
MWP-7	0.00	0.00	0.00	0.00	0.04	0.32	0.98	1.00	19.00	19.00
MWP-8	0.00	0.94	1.00	1.00	1.00	1.00	1.00	1.00	6.00	29.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	8.00	27.00
MWP-10	0.00	0.64	0.82	0.94	0.94	0.94	0.94	0.94	-	-
MWP-11	0.00	0.34	0.50	0.94	0.96	0.96	0.96	0.96	-	-
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	24.00
MWP-13	0.00	0.00	0.20	1.00	1.00	1.00	1.00	1.00	7.00	29.00
GFf-1	0.00	0.00	0.76	1.00	1.00	1.00	1.00	1.00	9.00	29.00

Table A.9: GFc-4, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	7.00	18.00
MWP-2	0.00	0.00	0.56	1.00	1.00	1.00	1.00	1.00	8.00	29.00
MWP-3	0.00	0.00	0.00	0.94	1.00	1.00	1.00	1.00	11.00	29.00
MWP-4	0.00	0.00	0.02	0.98	1.00	1.00	1.00	1.00	8.00	29.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	13.00
MWP-6	0.00	0.00	0.00	0.62	1.00	1.00	1.00	1.00	11.00	19.00
MWP-7	0.00	0.00	0.00	0.00	0.02	0.22	1.00	1.00	19.00	19.00
MWP-8	0.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	6.00	29.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	7.00	14.00
MWP-10	0.00	0.68	0.90	0.98	0.98	0.98	0.98	0.98	-	-
MWP-11	0.00	0.64	0.92	1.00	1.00	1.00	1.00	1.00	9.00	29.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	12.00
MWP-13	0.00	0.00	0.68	1.00	1.00	1.00	1.00	1.00	7.00	29.00
GFf-1	0.00	0.00	0.54	0.96	1.00	1.00	1.00	1.00	13.00	29.00

Table A.10: GFc-5, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	6.00	20.00
MWP-2	0.00	0.00	0.70	1.00	1.00	1.00	1.00	1.00	7.00	21.00
MWP-3	0.00	0.00	0.02	0.78	1.00	1.00	1.00	1.00	11.00	29.00
MWP-4	0.00	0.00	0.00	0.96	1.00	1.00	1.00	1.00	9.00	27.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	8.00	22.00
MWP-6	0.00	0.00	0.00	0.48	1.00	1.00	1.00	1.00	11.00	18.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.12	0.96	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	29.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	9.00	27.00
MWP-10	0.00	0.74	0.96	1.00	1.00	1.00	1.00	1.00	7.00	29.00
MWP-11	0.00	0.32	0.50	0.74	0.74	0.74	0.74	0.74	-	-
MWP-12	0.00	0.02	1.00	1.00	1.00	1.00	1.00	1.00	6.00	12.00
MWP-13	0.00	0.00	0.10	1.00	1.00	1.00	1.00	1.00	7.00	29.00
GFf-1	0.00	0.00	0.62	0.98	1.00	1.00	1.00	1.00	9.00	29.00

Table A.11: GFc-5, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	6.00	28.00
MWP-2	0.00	0.00	0.76	1.00	1.00	1.00	1.00	1.00	7.00	29.00
MWP-3	0.00	0.00	0.00	0.96	1.00	1.00	1.00	1.00	12.00	29.00
MWP-4	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	8.00	29.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	27.00
MWP-6	0.00	0.00	0.00	0.38	1.00	1.00	1.00	1.00	11.00	20.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.04	0.96	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	29.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	8.00	15.00
MWP-10	0.00	0.66	0.86	1.00	1.00	1.00	1.00	1.00	8.00	29.00
MWP-11	0.00	0.90	0.96	1.00	1.00	1.00	1.00	1.00	10.00	29.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	7.00
MWP-13	0.00	0.00	0.22	1.00	1.00	1.00	1.00	1.00	7.00	29.00
GFf-1	0.00	0.00	0.58	0.96	1.00	1.00	1.00	1.00	6.00	29.00

Table A.12: GFc-6, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.98	1.00	1.00	1.00	1.00	1.00	7.00	16.00
MWP-2	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	8.00	29.00
MWP-3	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	10.00	26.00
MWP-4	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	9.00	20.00
MWP-5	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	8.00	20.00
MWP-6	0.00	0.00	0.00	0.44	1.00	1.00	1.00	1.00	12.00	22.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.58	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	11.00
MWP-9	0.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	13.00	29.00
MWP-10	0.00	0.94	0.98	1.00	1.00	1.00	1.00	1.00	7.00	18.00
MWP-11	0.00	0.86	0.98	1.00	1.00	1.00	1.00	1.00	7.00	23.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	13.00
MWP-13	0.00	0.00	0.98	1.00	1.00	1.00	1.00	1.00	7.00	20.00
GFf-1	0.00	0.00	0.54	0.98	1.00	1.00	1.00	1.00	22.00	29.00

Table A.13: GFc-6, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	7.00	20.00
MWP-2	0.00	0.00	0.96	1.00	1.00	1.00	1.00	1.00	7.00	24.00
MWP-3	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	11.00	22.00
MWP-4	0.00	0.00	0.04	1.00	1.00	1.00	1.00	1.00	9.00	23.00
MWP-5	0.00	0.00	0.06	1.00	1.00	1.00	1.00	1.00	7.00	18.00
MWP-6	0.00	0.00	0.00	0.48	1.00	1.00	1.00	1.00	11.00	22.00
MWP-7	0.00	0.00	0.00	0.00	0.04	0.62	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	14.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	10.00	25.00
MWP-10	0.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	7.00	16.00
MWP-11	0.00	0.68	0.94	1.00	1.00	1.00	1.00	1.00	7.00	20.00
MWP-12	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	14.00
MWP-13	0.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00	7.00	18.00
GFf-1	0.00	0.00	0.64	1.00	1.00	1.00	1.00	1.00	6.00	29.00

Table A.14: GFc-7, only AIDEA, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.56	1.00	1.00	1.00	1.00	1.00	6.00	11.00
MWP-2	0.00	0.00	0.12	0.96	1.00	1.00	1.00	1.00	6.00	17.00
MWP-3	0.00	0.00	0.00	0.96	1.00	1.00	1.00	1.00	11.00	15.00
MWP-4	0.00	0.00	0.00	0.98	1.00	1.00	1.00	1.00	8.00	13.00
MWP-5	0.00	0.00	0.02	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-6	0.00	0.00	0.00	0.12	1.00	1.00	1.00	1.00	10.00	14.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.10	1.00	1.00	18.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	14.00	29.00
MWP-10	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	20.00
MWP-11	0.00	0.06	0.18	1.00	1.00	1.00	1.00	1.00	9.00	24.00
MWP-12	0.00	0.02	1.00	1.00	1.00	1.00	1.00	1.00	6.00	8.00
MWP-13	0.00	0.00	0.24	1.00	1.00	1.00	1.00	1.00	7.00	12.00
GFf-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	10.00	29.00

Table A.15: GFc-7, MP-AIDEA, 2 populations, $\delta_d = \delta_u = \delta_f = 0.1, \delta_c = 0$

	1000	5000	10000	50000	100000	200000	500000	1000000	$n_{iter,min}$	$n_{iter,max}$
MWP-1	0.00	0.00	0.60	1.00	1.00	1.00	1.00	1.00	7.00	14.00
MWP-2	0.00	0.00	0.14	1.00	1.00	1.00	1.00	1.00	11.00	23.00
MWP-3	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	11.00	16.00
MWP-4	0.00	0.00	0.00	0.94	1.00	1.00	1.00	1.00	10.00	18.00
MWP-5	0.00	0.00	0.00	1.00	1.00	1.00	1.00	1.00	7.00	9.00
MWP-6	0.00	0.00	0.00	0.22	1.00	1.00	1.00	1.00	11.00	16.00
MWP-7	0.00	0.00	0.00	0.00	0.00	0.18	1.00	1.00	19.00	19.00
MWP-8	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	6.00
MWP-9	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	18.00
MWP-10	0.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	6.00	18.00
MWP-11	0.00	0.08	0.14	1.00	1.00	1.00	1.00	1.00	9.00	27.00
MWP-12	0.00	0.02	1.00	1.00	1.00	1.00	1.00	1.00	6.00	11.00
MWP-13	0.00	0.00	0.26	1.00	1.00	1.00	1.00	1.00	7.00	10.00
GFf-1	0.00	0.00	0.82	1.00	1.00	1.00	1.00	1.00	6.00	29.00

Appendix A. Testing Results for the Single-Objective Min-Max Problem

Appendix A. Testing Results for the Single-Objective Min-Max Problem

Appendix B

Multi-Objective Min-Max Approach

It is often important to the decision maker, who is the final user of this methodology, to perform a trade-off analysis between the different QoIs. Sometimes the decision maker is able to define precisely a weight vector to rank by importance the QoIs. Sometimes, instead, this is not possible. The scalarisation approach is a well-known methodology to solve the MOP formulation by reducing it to a SOP. We prefer the scalarisation approach to the direct multi-objective optimisation because more flexible to be applied to both conditions in which objective weights are predefined or not. When the set of weights is given, the scalarisation approach finally reduces the computational cost.

This chapter presents a generalisation of the methodology proposed in Chapter 5 to CMOP min-max. We propose a Chebychev/Pascoletti-Serafini Scalarisation (CPSS), a combination of WCS [185] and PSS [186], to translate the CMOP in a SOP. We also suggest a generalisation of MP-AIDEA [131] to CMOP. A test case is analysed for which three different scalarisation approaches are compared: ECS, WSS [183] and CPSS where the last one is a smooth combination of WCS and PSS. A comparison of the different approaches can be found in [55].

This chapter is linked to Chapter 4 where it is given the definition of optimisation problems: the generic SOP, MOP, CSOP, CMOP, and in particular the min-max problem. Chapter 4 gives also the definition of the most used scalarisation approaches: WCS and PSS are here used and combined. This chapter is finally linked to Chapter 5 to which it represents a generalisation.

The rest of the chapter is organised as it follows. Appendix B.1 first presents the general problem formulation. Appendix B.2 explains the min-max method for the solution of the relaxation problem in Chapter 5. Appendix B.3 presents the generalisation of MP-AIDEA to CMOP. Appendix B.4 introduces to the test case. Appendix B.5 concludes the chapter.

B.1 Problem Formulation

The general problem we are interested to solve in this chapter is a CMOP min-max. The mathematical formulation Eq. (5.1) is here reported for clarity:

$$\begin{aligned} &\min_{\mathbf{d}\in D}\max_{\mathbf{u}\in U} f_i(\mathbf{d},\mathbf{u}) && \forall i \in I_f = [1,...,m]^T \\ &s.t. && \\ &c_i(\mathbf{d},\mathbf{u}) \leq 0 && \forall \mathbf{u}\in U, \forall i \in I_c = [1,...,s]^T \end{aligned} \tag{B.1}$$

We propose CPSS, a new scalarisation approach that combines WCS and PSS, to solve Eq. (B.1). The CMOP then can be translated to a CSOP:

$$\begin{aligned} &\min_{\mathbf{d}\in D}\left[\max_{\mathbf{u}\in U} \bar{f}^{\omega}(\mathbf{d},\mathbf{u})\right] && \\ &s.t. && \\ &c_i(\mathbf{d},\mathbf{u}) \leq 0 && \forall \mathbf{u}\in U, \forall i \in I_c = [1,...,s]^T \end{aligned} \tag{B.2}$$

in which $\bar{f}^{\omega}$ is the normalised and scalarised transformation of $\vec{f}$. Problem in Eq. (B.2) can then be solved with the method presented in Chapter 5.

This chapter focuses on a specific sub-class of Eq. (B.1) that arises as an alternative to the relaxation problem in Eq. (5.6) as explained in Fig. 5.1. The problem under analysis is then the unconstrained MOP min-max:

$$\min_{\mathbf{d}\in D}\left[\max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u}), \max_{\mathbf{u}\in U}\max_{i\in I_c} c_i(\mathbf{d},\mathbf{u})\right] \tag{B.3}$$

The constraint relaxation presented in Section 5.2.3 for the solution of Eq. (B.3) can be thought as a ECS approach [55, 182]. We suggest here instead a CPSS in order to be able to reconstruct the whole Pareto front for the bi-objective problem. Indeed, one could be interested in a trade-off between f and ϵ and accept larger relaxations of the constraints in favour of a better objective. Note that in the context of UQ this implies accepting a higher probability of violating the constraints in favour of a better cost function.

B.2 A Memetic Multi-Objective Alternative to the Relaxation Approach

We now introduce the assumption that a preference vector $\boldsymbol{\omega} = [\omega_f, \omega_c]^T$ can be defined a priori and explain the scalarisation approach summarised in Algorithm 15. By varying the values of $\boldsymbol{\omega}$ then the Pareto Front can be reconstructed.

In the first part (line 2 to 10) of Algorithm 15 the reference points c_{ideal}, f_{nadir}, f_{ideal} and c_{nadir} are calculated. c_{ideal} is the minimum (best) over D of the worst case

constraint violations in U:

$$c_{\text{ideal}} = \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} \max_{i\in I_c} c_i(\mathbf{d}, \mathbf{u}) \tag{B.4}$$

and it is equal to the relaxed constraint ϵ. For the corresponding design vector $\mathbf{d}_{c\text{-ideal}}$ the worst scenario for the objective function is f_{nadir} (lines 3 to 8):

$$f_{\text{nadir}} = \max_{\mathbf{u}\in U} f(\mathbf{d}_{c\text{-ideal}}, \mathbf{u}) \tag{B.5}$$

In line 9 the unconstrained min-max problem

$$f_{\text{ideal}} = \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d}, \mathbf{u}) \tag{B.6}$$

is solved to define the best design configuration $\mathbf{d}_{f\text{-ideal}}$ that minimises the worst scenarios of the objective function f regardless the constraint violation. $\mathbf{d}_{f\text{-ideal}}$ is then used in line 10 to calculate the corresponding worst condition for the constraint violation:

$$c_{\text{nadir}} = \max_{\mathbf{u}\in U} \max_{i\in I} c_i(\mathbf{d}_{f\text{-ideal}}, \mathbf{u}) \tag{B.7}$$

An example of the reference points for a generic Pareto front applied to the min-max problem is in Fig. B.1 where $\mathbf{z}^{\text{ideal}} = [f_{\text{ideal}}, c_{\text{ideal}}]$ and $\mathbf{z}^{\text{nadir}} = [f_{\text{nadir}}, c_{\text{nadir}}]$. In the second part (lines 13 to 27) it is instead described the scalarisation procedure. Nadir and ideal points are here used to normalise f and c:

$$\bar{f} = \frac{f - f_{\text{ideal}}}{f_{\text{nadir}} - f_{\text{ideal}}} \tag{B.8}$$

$$\bar{c} = \frac{\max_{i\in I_c} c_i - c_{\text{ideal}}}{c_{\text{nadir}} - c_{\text{ideal}}}. \tag{B.9}$$

Once the reference points are calculated, the algorithm follows a generalisation of what presented in Chapter 5 by alternating a minimisation and a restoration step. The minimisation step searches for a global solution to the constrained min-max problem:

$$\begin{aligned} &\min_{\mathbf{d}\in D} \max_{\mathbf{u}_{af}\in \bar{A}_{uf}} \bar{f}^{\omega}(\mathbf{d}, \mathbf{u}_{af}) \\ &s.t. \\ &\max_{\mathbf{u}_{ac}\in \bar{A}_{uc}} \max_{i\in I_c} c_i(\mathbf{d}, \mathbf{u}_{ac}) \leq 0. \end{aligned} \tag{B.10}$$

While the restoration step searches for a solution to the following two global maximisation problems, given the solution $\bar{\mathbf{d}}$ coming from Eq. (B.10):

$$\begin{aligned} &\max_{\mathbf{u}\in U} \bar{f}^{\omega}(\bar{\mathbf{d}}, \mathbf{u}) \\ &s.t. \\ &\max_{i\in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}) \leq 0 \end{aligned} \tag{B.11}$$

$$\max_{\mathbf{u}\in U} \max_{i\in I_c} c_i(\bar{\mathbf{d}}, \mathbf{u}) \tag{B.12}$$

The two archives $\bar{A}_{uf}$ and $\bar{A}_{uc}$ are defined as in Chapter 5.

The contribution given for the solution of Eq. (B.2) is included in the procedure for the solution of the single-layer problems in Eqs. (B.11) and (B.12) and it is explained in the following section.

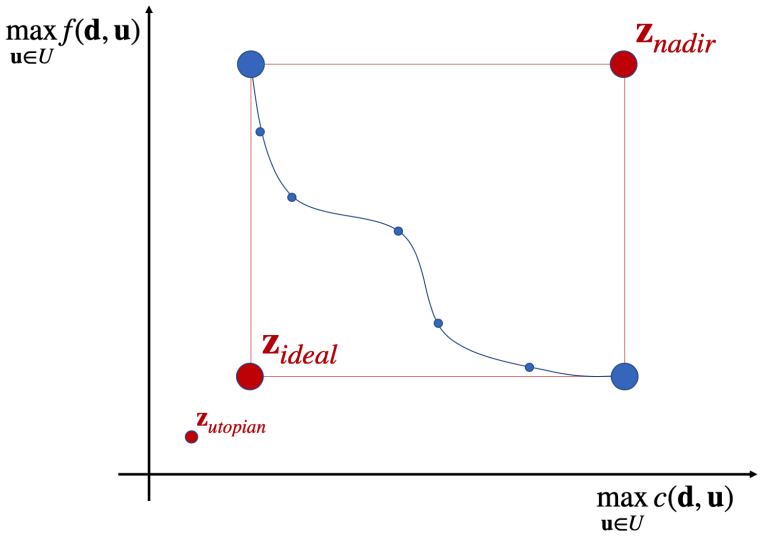


Figure B.1: A generic Pareto front for the min-max problem. In this case the functions $\max f$ and $\max c$ are considered as conflicting objectives. The ideal $\mathbf{z}_{\text{ideal}}$, nadir $\mathbf{z}_{\text{nadir}}$ and utopian $\mathbf{z}_{\text{utopian}}$ points are represented. They are theoretic points that collapse the extreme behaviour of the different solutions in the Pareto front. $\mathbf{z}_{ideal}$ is the combination of the best solutions for the different objectives. $\mathbf{z}_{\text{nadir}}$ represents instead the worst possible combination of points. $\mathbf{z}_{\text{utopian}}$ is finally defined by means of an ϵ from $\mathbf{z}_{\text{ideal}}$.

B.3 A Memetic Scalarisation Approach for Multi-Objective Optimisation

This section explains the novelties introduced to MP-AIDEA to solve the CMOP. The algorithm is presented in Algorithms 12 to 14. The contribution refers to the evaluation part of the candidate solutions and is twofold: within the DE step and within the local search part.

Within the DE, to translate the MOP to a single objective problem we propose to apply the WCS:

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{X}} \max_i\{\omega_i(f_i - z_{\text{ideal},i})\} & \forall i = 1,2,...,m \\ s.t. \quad c_j \leq 0 & \forall j = 1,2,...,n \end{array} \tag{B.13}$$

when the problem is not normalised, and

$$\begin{array}{ll} \min\limits_{\mathbf{x}\in\mathbb{X}} \max_i\{\omega_i \bar{f}_i\} & \forall i = 1,2,...,m \\ s.t. \quad c_j \leq 0 & \forall j = 1,2,...,n \end{array} \tag{B.14}$$

when it is normalised. During the local search, instead, the PSS is implemented because a differentiable fitness function is required. The following constrained minimisation problem is then considered

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X},t\in\mathbb{R}} \quad t \\ &s.t. \qquad\quad \omega_i(f_i - z_i) \leq t, \quad \forall i = 1,2,...,m \\ &\qquad\quad\; c_j \leq 0, \qquad\quad \forall j = 1,2,...,n \end{aligned} \tag{B.15}$$

when the problem is not normalised and

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{X},t\in\mathbb{R}} \quad t \\ &s.t. \qquad\quad \omega_i(\bar{f}_i - \bar{z}_i) \leq t, \quad \forall i = 1,2,...,m \\ &\qquad\quad\; c_j \leq 0, \qquad\quad \forall j = 1,2,...,n \end{aligned} \tag{B.16}$$

when it is normalised. In Eqs. (B.15) and (B.16) z_i ($\bar{z}_i$) is the best candidate solution f_i ($\bar{f}_i$) obtained in the previous DE. As stated in [187], Eqs. (B.15) and (B.16) could be considered as a reformulation (a linearisation) of the WCS where an additional variable is introduced and where the direction $r_i = 1/\omega_i$. However we consider here a different reference than the ideal point $\mathbf{z}_{\text{ideal}}$.

Algorithm 12 MP-AIDEA (Multi-Objective, Constraint handling)

1: Initialisation
2: **while** $n_{\text{feval}} < n_{\text{feval,max}}$ **do**
3: *Run the DE step* (Algorithm 13)
4: **for** $p \in [1,2,...,N_{\text{pop}}]$ **do**
5: **if** $\mathbf{x}_{p,\text{best}}$ not in the basin of attraction of previous solutions **then**
6: *Run local search* (Eq. (B.15)) with $\mathbf{x}_{0,p} = \mathbf{x}_{\text{best},p}$, $t_{0,p} = 0$ and the reference vector $\mathbf{z}_p$: $\min_{\mathbf{x}\in\mathbb{X},t\in\mathbb{R}} t$ s.t. $\omega_i(f_i(\mathbf{x}) - z_i) \leq t \wedge c_j(\mathbf{x}) \leq 0$, $\forall i = 1,2,...,m$, $\forall j = 1,2,...,n$
7: update $\mathbf{x}_{p,\text{best}}$ from the local search.
8: **end if**
9: **end for**
10: *Initialise* populations for local or global restart in the next DE step [131].
11: **end while**

B.4 Test Case

The approach is here tested to the combination of the objective function $GFf1$ in Table 5.1 and the constraint function $GFc1$ in Table 5.2 for the solution of the CMOP min-max and the reconstruction of the Pareto Front for the minimisation trade-off between the worst case scenarios $\max f$ and $\max \mathbf{c}$.

Three scalarisation procedures ECS, WSS and CPSS are used and compared. With reference to WSS, the minimisation over the design space D of the weighted sum of the

Algorithm 13 DE step (Multi-Objective, Constraint handling)

1: **for** $p \in [1, 2, ..., N_{\text{pop}}]$ **do**
2: *Initialise* (input) the genotype $\mathbf{x}_{p,q}^{(G)}$ for the p-population at generation $G = 1$ where $q = 1, 2, ..., n_{\text{pop}}$
3: *Evaluate* the phenotype of each candidate solution: $f_{s,p,q}^{(G)}$ (Algorithm 14)
4: **while** the population is not contracted **do**
5: *Select* parents: all generation G;
6: *Variate* the parent's genotype: two strategies randomly alternated (DE/Rand/1/bin, DE/CurrentToBest/2/bin) define generation G+1;
7: *Evaluate* new candidates $f_{s,p,q}^{(G+1)}$ (Algorithm 14):
8: *Select* between parents and children with a greedy criterion
9: update generation: G = G+1.
10: **end while**
11: $\mathbf{x}_{p,\text{best}} = \arg\min_i f_{s,p,q}^{(\text{end})}(\mathbf{x}_{p,q})$;
12: $\mathbf{z}_p = \{f_1(\mathbf{x}_{p,\text{best}}), f_2(\mathbf{x}_{p,\text{best}}), ..., f_m(\mathbf{x}_{p,\text{best}})\}$.
13: **end for**

Algorithm 14 DE Evaluation (Multi-Objective, Constraint handling)

1: **for** each q-agent in the population, with $q \in [1, 2, ..., n_{\text{pop}}]$ **do**
2: $f_{s,p,q} = \max_i\{\omega_i(f_i(\mathbf{x}_{p,q}) - z_{\text{ideal},i})\}$, $i \in [1, 2, ..., m]$
3: $c_{p,q} = \max_j\{c_j(\mathbf{x}_{p,q})\}$, $j \in [1, ..., n]$
4: **end for**
5: **for** each q-agent with $k \in [1, 2, ..., n_{\text{pop}}]$ **do**
6: **if** $c_{p,q} > 0$ **then**
7: $f_{s,p,q} = \max_i\{f_{s,p,q}\} + c_{p,q}$
8: **end if**
9: **end for**

worst case scenarios for f and c in the respective archives (line 18) of Algorithm 15 is:

$$\min_{\mathbf{d}\in D}\left[\omega_f \max_{\mathbf{u}\in\bar{A}_{uf}} \bar{f}(\mathbf{d},\mathbf{u}) + \omega_c \max_{\mathbf{u}\in\bar{A}_{uc}} \bar{c}(\mathbf{d},\mathbf{u})\right] \tag{B.17}$$

We used the modified version of MP-AIDEA with the following settings: maximum number of function evaluation $n_{\text{feval,max}} = 3000$, number of populations $n_{\text{pop}} = 2$, number of agents in the population $N_{\text{pop}} = 5$, dimension of the bubble for the global restart $\delta_{\text{global}} = 0.1$ and DE threshold for each population $\rho = 0.1$. The whole constrained min-max algorithm has then been run until convergence. For ECS the algorithm has been run for 60 different thresholds ranging between 1 and 6.2 (calculated minimum and maximum constraint violation). For WSS and CPSS, instead, the trigonometric weights

$$\begin{aligned} w_f &= \frac{\cos\theta}{\cos\theta+\sin\theta} \\ w_c &= \frac{\sin\theta}{\cos\theta+\sin\theta} \end{aligned} \tag{B.18}$$

have been used letting θ varying from 0 to $\frac{\pi}{2}$ and using a discretisation of 60 interval

as well.

The results are in Fig. B.2 which shows a comparison of the three methods. Fig. B.2 shows that WSS is not capable of finding optimal Pareto points in the non convex part of the Pareto front [213] and also, using equally spaced weights, it finds non-equally spaced points in the front. The ECS strategy gives better results. The best performance is, however, obtained with the CPSS. Indeed it allows the user to express a preference through the selection of the weights and, as shown in Fig. B.2, with the same number of simulations (60 different descent directions for the CPSS and epsilon values for the ECS) it is able to find more Pareto optimal solutions than ECS. For these reasons CPSS was chosen and implemented in Algorithm 2.

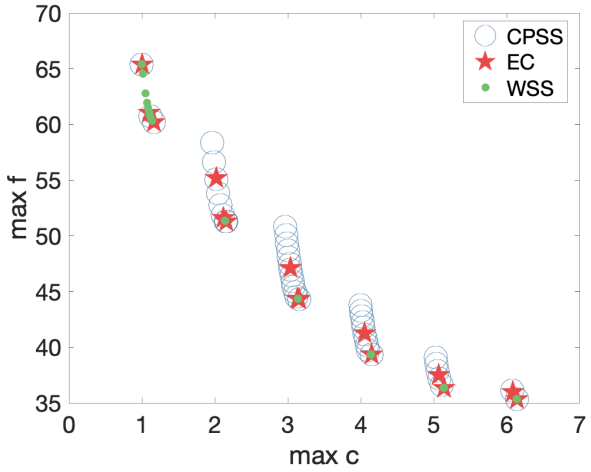


Figure B.2: Pareto front corresponding to the trade-off between the two conflicting goals $\max f$ and $\max c$. The results refer to test case $GFf1\&GFc1$. In particular, the ϵ-constraint (EC) approach has been applied with different thresholds while the Chebychev/Pascoleti Serafini (CPSS) and the Weighted-Sum (WSS) scalarisations have been applied with different preference vectors.

B.5 Conclusion

This chapter presented a methodology for the solution of CMOP min-max. We propose to reconduct the multi-objective formulation to a single-objective formulation through a scalarisation approach and in particular, with CPSS.

A particular sub-problem of CMOP min-max has been used to explain the methodology. The problem arises as an alternative to the relaxation method presented in Chapter 5. This approach is promising for cases in which a user-defined relaxation of the constraint is possible as it allows one to explore the optimal trade-off curve between optimality and reliability. The use of this approach will be further developed in future work.

Algorithm 15 Scalarisation Strategy

Inherit vectors from Algorithm 2
Normalisation points:
if Algorithm 2 not converged **then**
  run relaxation strategy in Algorithm 5
  $c_{\text{ideal}} = \epsilon = \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} \max_{i\in I_c} c_i(\mathbf{d},\mathbf{u})$
  $f_{\text{nadir}} = \max_{\mathbf{u}\in U} f(\mathbf{d}_{c\text{-ideal}},\mathbf{u})$
else
  c_{ideal} and f_{nadir} from Algorithm 2.
end if
run Algorithm 2 for the unconstrained problem on f:
$f_{\text{ideal}} = \min_{\mathbf{d}\in D} \max_{\mathbf{u}\in U} f(\mathbf{d},\mathbf{u})$
$c_{\text{nadir}} = \max_{\mathbf{u}\in U} \max_{i\in I} c_i(\mathbf{d}_{f\text{-ideal}},\mathbf{u})$
Scalarisation step:
while satisfy limits on $n_{fval}, n_{loop} \wedge$ not convergence **do**
  Minimisation loop:
  if Weighted-sum scalarisation **then**
    $\min_{\mathbf{d}\in D} \left[\omega_f \max_{\mathbf{u}\in \bar{A}_{uf}} \bar{f}(\mathbf{d},\mathbf{u}) + \omega_c \max_{\mathbf{u}\in \bar{A}_{uc}} \bar{c}(\mathbf{d},\mathbf{u})\right]$
    with the cross-check as in Algorithm 6,
    $\bar{f}$ and $\bar{c}$ defined in Eqs. (B.8) and (B.9)
  else if Chebyshev Pascoletti-Serafini scalarisation **then**
    during the Differential Evolution step:
    $\min_{\mathbf{d}\in D} \left\|\omega_f(\max_{\mathbf{u}\in A_{uf}} \bar{f}(\mathbf{d},\mathbf{u}), \omega_c \max_{\mathbf{u}\in A_{uc}} \bar{c}(\mathbf{d},\mathbf{u})\right\|_\infty$
    during the Local Search step:
    $\min_{\mathbf{d}\in D, t<0} t$
    s.t.
    $\omega_f\big(\max_{\mathbf{u}_{af}\in A_{uf}} \bar{f}(\mathbf{d},\mathbf{u}_{af}) - z_f\big) < t$
    $\omega_c\big(\max_{\mathbf{u}_{ac}\in A_{uc}} \bar{c}(\mathbf{d},\mathbf{u}_{uc}) - z_c\big) < t$
    $\bar{f}$ and $\bar{c}$ defined in Eqs. (B.8) and (B.9)
  end if
    Update global archive $A_d = A_d \cup \{\bar{\mathbf{d}}\}$
  Restoration loop:
    Run $\mathbf{u}_{af} = \arg\max_{\mathbf{u}\in\mathbf{U}} f(\bar{\mathbf{d}},\mathbf{u})$
    Run $\mathbf{u}_{ac} = \arg\max_{\mathbf{u}\in U} \max_{i\in I} c_i(\bar{\mathbf{d}},\mathbf{u})$
    if multiple outputs, choose best $\mathbf{u}_{ac}$: Algorithm 3
    Update archives: Algorithm 4
end while
for all $\mathbf{d} \in A_d$ from phase 2 **do**
  Cross-check: Algorithm 6
end for
Select which solution $[\mathbf{d}_{\text{opt}}, \mathbf{u}_{\text{opt}}]$ to return: Algorithm 7

Appendix B. Multi-Objective Min-Max Approach

Appendix B. Multi-Objective Min-Max Approach

Appendix C

Outer Belief Estimation via Evolutionary Binary Tree

The content of this chapter was published in:

- C. O. Absil, M. Vasile, **G. Filippi**, A. Riccardi, and M. Vasile, A Variance-Based Estimation of the Resilience Indices in the Preliminary Design Optimisation of Engineering Systems Under Epistemic Uncertainty, in *EUROGEN, (Madrid)*, 2017 [60]

The chapter presents the results of the work published in the paper above for which the main author is Carlos Ortegga and I partially contributed.

For an exact reconstruction of the Belief (resp. Plausibility) curve, the determination of the worst event (resp. best-case event) is necessary over every subset of the uncertainty space that has a non-null bpa. In the general case, this translates into a number of global maximisations (resp. minimisations) of the quantity of interest $F(\mathbf{u})$. This section will focus on the estimation of the Belief curve

$$Bel(F(\mathbf{u}) \leq \nu) = \sum_{\overline{\Omega}} m(\Omega)\,,$$
$$\overline{\Omega} = \{\Omega \subset \Theta | \max_{\mathbf{u} \in H_1 \subseteq \Omega} (F(\mathbf{u})) \leq \nu\}$$

of a design over all possible values of ν. Note the exact computation of the entire curve can be conducted by a cumulative sum of *bpa* over the sorted maxima

$$\mathcal{F} = \{ \max_{\mathbf{u} \in H_1 \subseteq \Omega} F(\mathbf{u})\,,\ \Omega \subset \Theta\,,\ m(\Omega) > 0\}\,.$$

The extension to the calculation of Plausibility is immediate, and the ideas exposed hereby can easily be applied to the computation of propositions stated otherwise.

In the algorithm proposed, the whole computation of Belief proceeds by building a tree that has at its root the whole uncertainty space with the associated global worst-case optimisation solution, and at its terminal nodes the whole set of focal elements,

each one with an associated maximum of the quantity of interest. The heuristic that drives how the tree is built and explored is key to the rapid convergence of the correct Belief and Plausibility values. The overall procedure is schemed in Algorithm 16 and detailed in the following subsections.

C.1 Truncated Estimation

The truncated estimation process begins with a global maximisation over the whole uncertainty space U as a zeroth iteration

$$S_0 := U\,,\ \bar{F}_0 = \max_{\mathbf{u}\in S_0^0} F(\mathbf{u})\,.$$

This allows one to assert that

$$\begin{aligned} Bel(F(\mathbf{u}) \leq \nu) = 1 \quad \nu \geq \bar{F}_0\,,\\ Bel(F(\mathbf{u}) \leq \nu) \geq 0 \quad \nu < \bar{F}_0\,, \end{aligned}$$

and is equivalent to propagation of the vacuous Belief function [50] to quantity F over U.

Then a s-subdivision of the search space is proposed,

$$S_1^1 \cup S_1^2 \cup \cdots \cup S_1^s = S_0^0\,,$$

where s is a hyper-parameter of the process. Since this split happens recursively, at iteration $i \geq 1$ one has a set

$$S_i = \{S_i^k\,,\ 1 \leq k \leq s^i\}$$

of subsets under consideration. Global optimisation is used to obtain the set of maxima

$$\bar{F}_i = \{\bar{F}_i^k = \max_{\mathbf{u}\in S_i^k} F(\mathbf{u})\,,\ 1 \leq k \leq s^i\}\,.$$

Let us assume for the sake of simplicity that k is redefined here so that the maxima are sorted $\bar{F}_i^k \leq \bar{F}_i^{k+1}$. Then it stands

$$\mathbf{u} \in \bigcup_{\kappa=1}^{k} S_i^\kappa \Longrightarrow F(\mathbf{u}) \leq \bar{F}_i^k$$

which allows one to compute the s^i-truncated approximation of the Belief curve

$$\widetilde{Bel}(F(\mathbf{u}) \leq \nu) = \sum_{\widetilde{\Omega}} m(\Omega)\,,$$

$$\widetilde{\Omega} = \{\Omega \subseteq \bigcup_{\kappa=1}^{k} S_i^\kappa | \bar{F}_i^k \leq \nu\,\}$$

by cumulative sum of *bpa* over $\bar{F}_i$. This sum can usually be simplified by considering degenerate *bpa* structures [112]. Such approximation is conservative by construction, i.e.

$$Bel(F(\mathbf{u}) \leq \nu) \geq \widetilde{Bel}(F(\mathbf{u}) \leq \nu)\,,$$

If all the domain is plausible, i.e. if

$$U = \bigcup \Omega\,,\ \Omega \subset \Theta\,,\ m(\Omega) > 0\,,$$

then

$$Bel(F(\mathbf{u}) \leq \nu) = \widetilde{Bel}(F(\mathbf{u}) \leq \nu) = 1 \Longleftrightarrow \nu \geq \bar{F}_0\,.$$

Furthermore, if it holds that

$$\mathbf{u} \in H_\mathbf{l} \subseteq \bigcup_{\kappa=1}^{k} S_i^\kappa \Longleftrightarrow \max_{\mathbf{u}\in H_\mathbf{l}}(F(\mathbf{u})) \leq \bar{F}_i^k$$

then

$$Bel(F(\mathbf{u}) \leq \bar{F}_i^k) = \widetilde{Bel}(F(\mathbf{u}) \leq \bar{F}_i^k)\,.$$

Assuming the exactitude of the global optimisation, it is clear that $\bar{F}_i \subset \bar{F}_{i+1}$, it will nonetheless be assumed that it is necessary to repeat these optimisations; this assumption will help contain the computational cost and is coherent with the conservative-approximation objective of this work. Thus, the cost of the overall process running for $0 \leq i < i_{max}$ is at most $\frac{s^{imax}-1}{s-1}$ global maximisations. Up to this point, the partitioning algorithm is equivalent to the binary tree approach proposed in [214] with the exception that at each level i more than two subsets are possible. The critical point is now to properly partition the U space and its subsets at every level i.

C.2 Heuristics for Minimisation of the Error

Even if we limit the domain subdivision procedure to a simple heuristic such as division over all the intervals of one and only one uncertain variable for the whole iteration, there are $\delta!$ possible variations of the aforementioned tree, where $\delta \leq dim(U)$ is the number of uncertain variables that have more than one interval with non-zero bpa.

The algorithm proposed in this paper stores proposes a breadth-first approach in successive iterations. It stores in an archive A_S the pairs $(\mathbf{u}, F(\mathbf{u}))$ evaluated by the optimisation process every time it is run over a subdomain S to compute

$$\max_{\mathbf{u}\in S\subseteq U} F(\mathbf{u})\,.$$

The archive can include information of previous iterations too. In the case that a deterministic optimisation algorithm is employed, a pre-sample of U can be used to increase the information available during the first iterations.

After the optimisation, A_S is used to decide on a s-subdivision the current space.

Appendix C. Outer Belief Estimation via Evolutionary Binary Tree

A function σ is defined

$$\sigma : A_S \longrightarrow \{S^1, S^2, \cdots, S^s\}\,,\ S^1 \cup S^2 \cup \cdots \cup S^s = S\,.$$

An appropriate choice of σ will lead to the construction of a tree such that it can be truncated at the desired depth with minimum approximation error of the Belief curve.

The heuristics proposed hereby will consider dividing S along one direction of uncertainty u_j at a time. Furthermore, it will be considered that

$$S^k = \bigcup (H_\mathbf{l} \,|\, H_\mathbf{l} \subset S \wedge l_j = k)\,,$$

which is equivalent to subdivide S along all intervals $[a_l, b_l] \mid bpa_{l,j} > 0$ for one of the non-singleton variables $u_j \,|\, L_j > 1$.

The statements in C.1 stand by considering

$$s \geq \max_j(L_j)\,.$$

This will from now on be referred to as *breadth-first* exploration of the truncated estimation tree.

Under such premises, defining σ reduces to selecting the direction u_j along which next split will take place. Since the truncated estimation is conservative by construction, σ is chosen hereby so to partition S along the direction that, according to A_S, captures the highest variability of the system budget with respect to the worst case in S. The idea of systematic partition along the u_j by sensitivity analysis on F is introduced in [112] . Here we consider, for each non-singleton u_j of S, the list of maxima

$$\widetilde{F}_j = \{\widetilde{F}_j^k = \max_{\substack{(\mathbf{u}, F(\mathbf{u})) \in A_S \\ \mathbf{u} \in S_j^k}} F(\mathbf{u})\,,\ 1 \leq k \leq L_j\}\,,$$

which constitutes a prediction of the next-iteration maxima in S if that direction is selected for subdivision. Let us assume once again that k is redefined so that the maxima are sorted $\widetilde{F}_j^k \leq \widetilde{F}_j^{k+1}$. The direction selected will then be

$$j = \arg\max_j \frac{\sum_{k=1}^{L_j - 1} (\widetilde{F}_j^{L_j} - \widetilde{F}_j^k)^2}{L_j - 1}\,,$$

which, by analogy with a variance measure, gives the *variance-based* designation.

This heuristics is designed as to favour a desirable estimated Belief curve over S, i.e. one that grows slowly in the high robustness values. If the maxima in $\widetilde{F}$ constitute a good approximation of the actual maxima over the S^k, which will be the case if the global optimiser explored S effectively, this will compensate the conservative approximation of the truncated estimation. Otherwise, the possible effects of under-exploration of some regions during the previous global maximisations will be mitigated for subsequent iterations.

It is noteworthy that the selection does not account for the *bpa* distribution among

Algorithm 16 Variance-based breadth-first reconstruction of the truncated Belief curve

Initialise $S = U$, $S_0 = \{S\}$ and $i = 0$
while $i < i_{max}$ **do**
  $\bar{F} \leftarrow \{\emptyset\}$, $S_{i+1} \leftarrow \{\emptyset\}$
  for $S \in S_i$ **do**
    $\bar{F} \leftarrow \bar{F} \cup \{\max_{\mathbf{u} \in S} F(\mathbf{u})\}$
    $A_S \leftarrow$ global optimisation history sample
    $S_{i+1} \leftarrow S_{i+1} \cup \sigma(A_S)$
  end for
  Reconstruct $\widetilde{Bel}$ curve from sorted($\bar{F}$) and *bpa*
  Apply termination condition if any
  $i \leftarrow i + 1$
end while
Return Last $\widetilde{Bel}$ curve

the S^k. If subdivisions can be selected that are very heterogeneous in *bpa*, then other σ options are preferable for a fast convergence of $\widetilde{Bel}(F \leq \nu)$ to $Bel(F \leq \nu)$. In fact, a strongly uneven distribution of bpa's can lead to substantial underestimations of the Belief if only the pure variance of the maxima is considered. The authors propose for instance maximising the area under the next-iteration prediction of the overall curve.

Note also that the purpose is here to obtain a good approximation of the Belief curve at a given cost. The designer might be interested in a higher detail for the pessimistic cases, for example, or be only interested in $Bel(F \leq \nu)$ for a given ν; then one should explore the tree otherwise than breadth-first. Combining the ideas exposed in [214] with a $\sigma(A_S)$ subdivision function to accelerate convergence will be the focus of future research.

Apart from the discussion on using the subdivision selection function σ as a heuristic to accelerate convergence to the exact Belief curve instead of a more traditional sensitivity analysis, the model reduction methodology of Algorithm 16 can be assimilated to a generalisation to arbitrary partitions of the step-wise construction algorithm proposed in [215] . There are however two more core novelties. First, the partitions considered are not obtained systematically per variable after an initial sensitivity analysis, but selected on-line, which allows the algorithm to better represent the coupling between sensitivities when exploring different regions of U. Secondly, the available data of a global optimisation history over each subspace visited is included in the analysis.

C.2.1 Evolutionary Binary Tree Application

The method is here applied and verified in the preliminary reliability-based design of the solar array of a small spacecraft.

The Problem

This section presents the application of the algorithm presented to the reliability-based sizing of the solar array of a small spacecraft power system, to be optimised in terms of construction cost and total power-generating surface. Three different formulations of increasing complexity will be proposed in the following sections, where each one is a particular case of the next. In all of them, a design will consist on a certain choice of the quantities:

- $A \in [A_{min}, A_{max}]$, the power-generating surface of the solar panel, [m^2].
- $\mu \in [0, 1]$, defines the proportion of cells of type I used in the solar panel. Each type, I and II, has its:
  - Best and worst-case solar efficiencies and failure profile, modelled as expert-provided probability assignments to efficiency intervals.
  - Cost per square meter of power-generating surface.

Hence the construction cost of a design $C(\mu, A)$ can be computed independently of the uncertainties. The sources of risk are, besides the solar cell efficiencies:

- Uncertainty on the power consumption of each of the subsystems, mostly due to lack of definition of the exact mission requirements, modelled as expert-provided probability assignments to power requirement intervals.
- Uncertainty on the power generation, mostly due to sparse background data on components recently adopted by the satellite provider, modelled as expert-provided probability assignments to power efficiency intervals.

The model considers 11 power consumptions of low design margin defined over an only interval, 14 power consumptions of high design margin defined over two intervals with distinct probability assignments, and 6 efficiency power ratios also defined over two intervals. Hence $\dim(U) = 31$ and there are $\delta = 20$ non-singleton directions u_j. The uncertainty space is composed of 2^{20} adjacent focal elements.

Note that it is the epistemic uncertainty that is predominant in all cases at this stage of the design. Hence all uncertainties have been modelled as epistemic and will be propagated through the system model by means of Evidence Theory. The reliability index selected is:

$$Bel(P_{gen} \geq P_{req})$$

Where P_{gen} is the power generated by the solar array and P_{req} is the power level required by the system, both uncertain. Thus, $Bel(P_{gen} \geq P_{req})$ is the most conservative probability estimation associated to the event of *satisfying the power requirements of the system* that can be inferred from the available evidence, and one will be interested in its maximisation or equivalently in the minimisation of the risk index

$$Pl(P_{gen} < P_{req})\,.$$

Risk Assessment of the Worst-Case Solution

With this formulation the worst-case-scenario optimum d^*_{wcs} is sought for the construction cost $C(\mu, A)$ by solving the problem:

$$\min_{d \in D} C(\mu, A)$$

$$s.t. : P_{gen} \geq P_{req} \quad \forall \mathbf{u} \in U$$

This is equivalent to requesting from the system a reliability index of 100% or risk index of 0% an can be solved analytically in this case by fixing $A | P_{gen} \geq P_{req} \quad \forall u \in U$ and minimising over μ.

For this design solution, the proportion of solar cells of each type μ^*_{wcs} is fixed and a risk analysis is then conducted varying the power-generating surface A. The reliability index

$$Bel(P_{gen} \geq P_{req}) = 1 - Pl(P_{gen} < Preq)$$

is presented against $C(\mu^*_{wcs}, A)$. This curve is estimated within 11 subsequent iterations of the variance-based algorithm proposed hereby. Since the maximisations are analytical over any subset of focal elements considered, a global optimisation log is not available and is hence mimicked with an initial latin hypersquare sample of cardinality 64. The curves thus obtained are compared to the exact curve computed in an exhaustive fashion requiring maximisation over all the focal elements of uncertainty – 2^{20} analytical maximisations in this case.

By construction this curve acts as a lower bound for the maximum reliability index of any design d^* that lays in the risk-budget Pareto set. In other words, it constitutes a lower bound to the overall reliability-budget trade-off curve whose computation is presented in C.2.1. Besides, its rightmost point corresponds to the worst case of the worst-case optimum and is thus assured to belong to the risk-budget trade-off Pareto front.

Bi-Objective Formulation

With this formulation the computation of the whole risk-budget trade-off Pareto front is tackled for the construction cost of the solar array $C(\mu, A)$. This can be expressed as:

$$\min_{d \in D} \begin{cases} C(\mu, A) \\ Pl(P_{gen} < P_{req}) \end{cases}$$

This is analogous to solving the family of evidence-based reliability-constrained optimisation problems

$$\min C(\mu, A)$$

$$s.t. : Pl(P_{gen} < P_{req}) \leq \epsilon$$

$$\epsilon \in [0, 1]$$

The problem above is solved by means of a single run of the multi-objective optimi-

sation algorithm Multi-Agent Collaborative Search (Multi Agent Collaborative Search (MACS) [216]), using 7 iterations of the variance-based algorithm proposed hereby for the approximation of the risk index at each function evaluation. No additional heuristics are added. Note that this formulation is as of today practically intractable without an approximation method for the risk index even for a problem that allows analytical maximisation over the focal elements, since it would require global optimisation over the design space on top of the exhaustive computation of the index over all the focal elements of uncertainty.

Three-Objective Formulation

With this formulation the computation of the whole risk-budget trade-off Pareto front is tackled for the construction cost of the solar array and its power-generating surface simultaneously. This can be expressed as

$$\min_{d\in D} \begin{cases} C(\mu, A) \\ A \\ Pl(P_{gen} < P_{req}) \end{cases}$$

and is analogous to solving the family of bi-objective evidence-based reliability-constrained optimisation problems

$$\min_{d\in D} \begin{cases} C(\mu, A) \\ A \end{cases}$$
$$s.t. : Pl(P_{gen} < P_{req}) \leq \epsilon$$
$$\epsilon \in [0, 1]$$

The solar array cells are such that type II have lower construction cost per kW of power generated but require higher power-generating surface. This holds both when comparing each type's best-case and worst-case parameters. Hence it is expected to find designs with $\mu = 0$ and $\mu = 1$ at the minimal-budget and minimal-surface extrema of the Pareto front, respectively. This problem is solved with the same set-up described in C.2.1.

C.2.2 Results

Risk Assessment of the Worst-Case Solution

Figure C.1 shows the increasing quality of the estimations obtained in 11 successive iterations of the variance-based approximation algorithm. The convergence to the exact curve on the conservative side is assured by construction of the algorithm, but it is still noteworthy that in this case the convergence rate is large enough as to obtain more precision than is necessary for the purposes that occupy the designer, while achieving a reduction of four orders of magnitude in the computational cost (wrt its exact computation). The heuristics used constitute a model reduction technique in the sense that they compile information represented along some directions of uncertainty, deemed less relevant. Hence these results are not generic, but the convergence speed will be

directly related to the reducibility properties of the index to estimate with respect to the problem uncertain variables in a given probability segment. In other words, the maximum estimation error will be obtained when the effect of every uncertain variable is homogeneous and there is no partitioning more significant than another amongst the considered. The problem defined hereby is found to be dominated by the effect of the uncertainty defined on the 6 power efficiencies, of which only the 3 of them corresponding to cells of type II are relevant hereby. Thus 7 iterations of the algorithm are henceforth deemed sufficient to capture most variability.

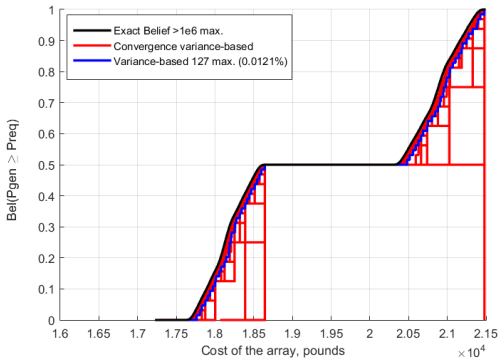


Figure C.1: Progressive approximation of the exact reliability-budget curve of a design solution composed entirely of cells of type II ($\mu = 0$). Highlighted, the approximation corresponding to 7 iterations yielding 127 maximisations, i.e. 0.0121% of the computational cost of obtaining the exact curve.

Bi-Objective Formulation

Figure C.2 illustrates the Reliability Pareto Front obtained for the problem in its bi-objective formulation. As discussed in section C.2.1, the curve in figure C.1 constitutes a lower bound for the complete reliability-budget trade-off curve and its rightmost point is coincident. In this particular case, since cells of type II have lower construction cost per kW of power generated both in the best and worst case, the leftmost point is also coincident. In this situation one could expect the front to be completely coincident, nevertheless the results show that solving the evidence-based reliability-constraint optimisation problem with a requirement in the reliability index between 0.5 and 0.75 would lead to optimal solutions composed by around 50% of cells of each type.

Figure C.3 proves that the reliability-budget curve varying A of a solution with $\mu = 0.5$ (dashed line) has both a best case and worst case suboptimal to those of a solution with $\mu = 0$ (solid line), but the former presents two plateaus instead of one and offers thus a higher lower bound on the cdf of the system at a lower construction cost in this reliability range. In this case the designers are more interested in the upper range of reliability and might focus their interest in the budget difference between worst-case cost, nonetheless the availability of this information provides a powerful decision-making tool in a generic scenario.

It can be noted that the quality of the approximation is worse for the design with

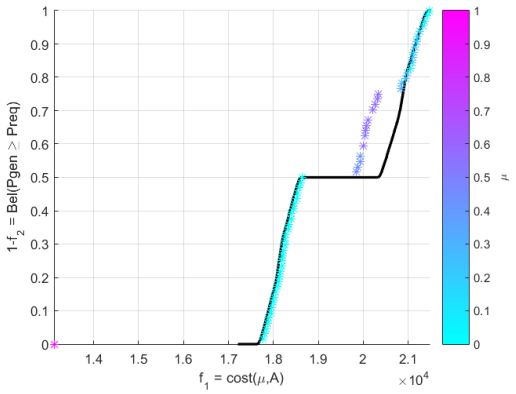


Figure C.2: Reliability Pareto front obtained for the design problem in its bi-objective formulation superposed to the exact Belief curve of the worst-case optimum (black line). Colours relate to the proportion of cells of type I and II.

mixed types of cells using the same estimation set-up. This is due to the fact that each of the maxima used to reconstruct the curve captures, in 7 iterations, the information as divided along 6 of the $\delta = 20$ non-singleton directions of epistemic uncertainty defined. For a design with $\mu = 0.5$, the indices will be more or less equally sensitive to the uncertainty in the parameters of cells of type I and type II, resulting in an homogenisation of the problem landscape. The algorithm, forced to account for more cell-type-related parameters, generates less or no subdivision along the subspace of U corresponding to the uncertainties in the power consumption, resulting in lower detail. Despite this fact, it has been shown that the quality of the approximation is enough for the bi-objective approach to spot the different behaviour of the solutions and attract attention towards a potentially interesting mixed-type solution.

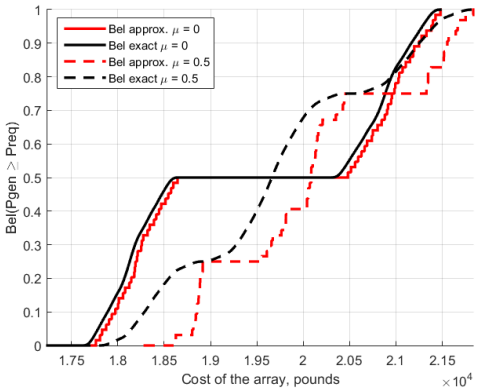


Figure C.3: Reliability-budget curves of design solutions with $\mu = 0$ and $\mu = 0.5$. Both the exact curves and those obtained with 7 iterations of the variance-based estimation algorithm are shown.

Three-Objective Formulation

Figure C.4 shows the family of optimal-budget Pareto fronts obtained in the three-objective formulation for every possible level of reliability requested from the design solution. Of course the fronts with a higher reliability associated are dominated by those that allow a higher risk index. The uppermost front corresponds to the worst-case Pareto optimal solutions. This front can be obtained at a reduced cost using multi-objective worst-case optimisation heuristics such as the ones integrated in MAC-Sminmax. Note that, whereas low and high-reliability solutions constitute almost-linear fronts in the budget space, requesting reliability values between 25 and 75% will lead to more exotically shaped Pareto fronts. In particular, the front becomes non-convex under 50% reliability index, indicating an abrupt change in the properties of the problem landscape. Figure C.5 presents the exact same information in a three-dimensional fashion, plus colours relate to the proportion of cells of type I and II used. It can be observed that, as predicted, there is one type of cell that will generally lead to reduction of the cost whereas the other will lead to reduction of the solar array power-generating surface.

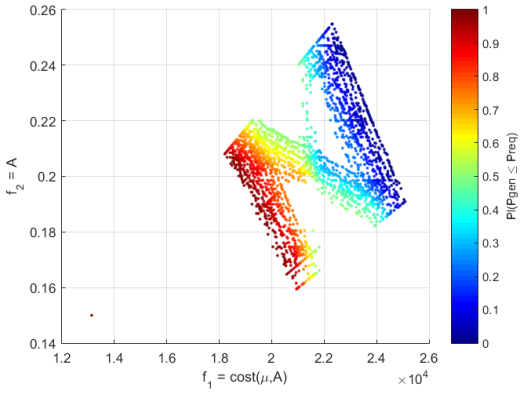


Figure C.4: Reliability-budget Pareto Front obtained for the design problem in its three-objective formulation projected to the budget axis, colours relate to the reliability index.

C.2.3 Conclusions

A methodology has been presented for the fast and conservative estimation of the Belief and Plausibility curves associated to a system budget of quantity of interest. This finds application in Evidence-Theoretic Uncertainty Quantification. The proposed algorithm relies on breadth-first partitioning of the uncertain space and model reduction after analysis of data coming from a global optimisation history. Heuristics to relate the partitioning scheme to the optimisation archive have been proposed and discussed.

The overall procedure has been put to the test by means of application in the Expert-Based Reliability Design Optimisation of the solar array of a small spacecraft. For this application, several formulations are proposed of increasing computational complexity

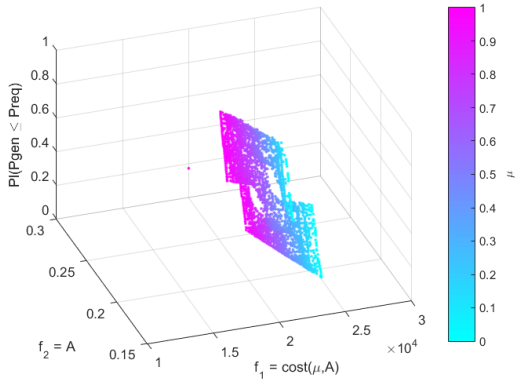


Figure C.5: Risk-budgets Pareto front obtained for the design problem in its three-objective formulation, colours relate to the proportion of cells of type I and II.

to obtain reliability-budget trade-off solutions. Such a detailed analysis is as of today intractable in large-scale engineering problems without a suitable approximation method for the system indices, even if simplified models are in use.

The results show that the proposed methodology can, under suitable model reduction assumptions, provide a large cut-off in computational cost with respect to the exact computation of the Belief and Plausibility curves, while maintaining a minimal approximation error.

It is nonetheless noteworthy that to tackle some of the formulations presented, namely those that involve multi-objective optimisation, only a value of the curve is of interest to drive the search. A preliminary discussion on heuristics to further reduce the cost in such applications has been led. A broader view on efficient robustness and reliability optimisation algorithms will constitute the focus of future research.

Appendix C. Outer Belief Estimation via Evolutionary Binary Tree

Bibliography

[1] V. Politi, "Scientific revolutions, specialization and the discovery of the structure of dna: toward a new picture of the development of the sciences," *Synthese*, vol. 195, pp. 2267–2293, 2018.

[2] W. Weaver, "Science and complexity," *American scientist*, vol. 36, no. 4, pp. 536–544, 1948.

[3] A. Vespignani, "L'algoritmo e l'oracolo. Come la scienza predice il futuro e ci aiuta a cambiarlo.," Il Saggiatore, 2019.

[4] C. N. Calvano and P. John, "Systems engineering in an age of complexity," *Systems Engineering*, vol. 7, pp. 25–34, jan 2004.

[5] F. J. Kahlen, S. Flumerfelt, and A. Alves, "Transdisciplinary perspectives on complex systems: New findings and approaches," *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, pp. 1–327, 1 2016. Mathematical Characterization of System-of-Systems Attributes.

[6] S. Lloyd, "Measures of complexity: A nonexhaustive list difficulty of description," 2001.

[7] M. Mitchell, "Complexity: A guided tour," in *Oxford University Press*, 2017.

[8] D. J. Snowden and M. E. Boone, "A leader's framework for decision making," *Harvard Business Review Home*.

[9] S. A. Sheard and A. Mostashari, "Principles of complex systems for systems engineering," *Systems Engineering*, vol. 12, pp. 295–311, sep 2009.

[10] P. Pedersen and C. L. Laureen, "Design for Minimum Stress Concentration by Finite Elements and Linear Programming," *Journal of Structural Mechanics*, vol. 10, pp. 375–391, jan 1982.

[11] H. G. Beyer and B. Sendhoff, "Robust optimization - A comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33-34, pp. 3190–3218, 2007.

[12] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering and System Safety*, vol. 94, no. 2, pp. 125–141, 2009.

[13] G. Filippi, M. Vasile, D. Krpelik, P. Z. Korondi, M. Marchi, and C. Poloni, "Space systems resilience optimisation under epistemic uncertainty," *Acta Astronautica*, vol. 165, pp. 195–210, dec 2019.

[14] N. V. Sahinidis, "Optimization under uncertainty: State-of-the-art and opportunities," in *Computers and Chemical Engineering*, vol. 28, pp. 971–983, Pergamon, jun 2004.

[15] M. Nicolich and G. Cassio, "System models simulation process manangement and collaborative multidisciplinary optimization," in *CEUR Workshop Proceedings*, vol. 1300, (Rome, Italy), pp. 1–16, 2014.

[16] X. Zhuang, R. Pan, and L. Wang, "Robustness and reliability consideration in product design optimization under uncertainty," in *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1325–1329, IEEE, dec 2011.

[17] A. Clarich, C. Poloni, and V. Pediroda, "A Competitive Game Approach for Multi Objective Robust Design Optimization," in *AIAA 1st Intelligent Systems Technical Conference*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, sep 2004.

[18] V. Pediroda, C. Poloni, and A. Clarich, "A fast and robust adaptive methodology for design under uncertainties based on dace response surface and game theory," pp. 29–36, 2005.

[19] G. J. Park, T. H. Lee, K. H. Lee, and K. H. Hwang, "Robust design: An overview," jan 2006.

[20] X. Du and W. Chen, "Towards a better understanding of modeling feasibility robustness in engineering design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 122, no. 4, pp. 385–394, 2000.

[21] L. Padovan, V. Pediroda, and C. Poloni, "Multi Objective Robust Design Optimization of Airfoils in Transonic Field," pp. 283–295, Springer, Berlin, Heidelberg, 2005.

[22] P. Crucitti, V. Latora, and M. Marchiori, "Model for cascading failures in complex networks," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 69, no. 4, p. 4, 2004.

[23] G. Punzo, A. Tewari, E. Butans, M. Vasile, A. Purvis, M. Mayfield, and L. Varga, "Engineering Resilient Complex Systems: The Necessary Shift Toward Complexity Science," *IEEE Systems Journal*, vol. PP, pp. 1–10, 2020.

[24] G. P. Cimellaro, A. M. Reinhorn, and M. Bruneau, "Framework for analytical quantification of disaster resilience," *Engineering Structures*, vol. 32, no. 11, pp. 3639–3649, 2010.

[25] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience," *Reliability Engineering and System Safety*, vol. 145, pp. 47–61, jan 2016.

[26] D. D. Woods and E. Hollnagel, *Essential characterstics of resilience*. 2006.

[27] M. Ruth, S. Goessling-Reisemann, and D. D. Woods, "Essentials of resilience, revisited," *Handbook on Resilience of Socio-Technical Systems*, no. January, pp. 52–65, 2019.

[28] S. N. Naghshbandi, L. Varga, A. Purvis, R. McWilliam, E. Minisci, M. Vasile, M. Troffaes, T. Sedighi, W. Guo, E. Manley, and D. H. Jones, "A review of methods to study resilience of complex engineering and engineered systems," *IEEE Access*, vol. 8, pp. 87775–87799, 2020.

[29] Lockheed Martin Corporation, "Guide to the Systems Engineering Body of Knowledge – G2SEBoK," *Guide to the Systems Engineering Body of Knowledge (SEBoK)*, p. 945, 2004.

[30] J. M. Epstein, "Why Model?," in *Journal of Artificial Societies and Social Simulation*, vol. 11, p. 12, 2008.

[31] W. J. Wertz, James R and Larson, *Space Mission Analysis and Design*. Kluwer Academic Publishers, space tech ed., 1999.

[32] M. Locatelli and F. Schoen, "(Global) Optimization: Historical notes and recent developments," *EURO Journal on Computational Optimization*, vol. 9, p. 100012, 2021.

[33] M. Locatelli and F. Schoen, *Global Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2013.

[34] A. H. A. Riccardi, E. Minisci, K. Akartunal, C. Greco, N. Rutledge, A. Kershaw, "Introduction to Optimisation," pp. 1–151, 2019.

[35] J. Stork, A. E. Eiben, and T. Bartz-Beielstein, "A new taxonomy of global optimization algorithms," jun 2022.

[36] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003.

[37] I. Rahimi, . Amir, H. Gandomi, . F. Chen, and E. Mezura-Montes, "A review on constraint handling techniques for population-based algorithms: from single-objective to multi-objective optimization," vol. 30, pp. 2181–2209, 2023.

[38] A. R. Jordehi, "A review on constraint handling strategies in particle swarm optimisation," *Neural Comput & Applic*, 2015.

[39] A. E. Smith and D. W. Coit, "Penalty functions," in *Handbook of Evolutionary Computation* (T. Baeck, D. Fogel, and Z. Michalewicz, eds.), ch. C 5.2, Oxford University Press, Institute of Physics Publishing, 1995.

[40] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA JOURNAL*, vol. 51, 2013.

[41] K. Shimizu and E. Aiyoshi, "Necessary conditions for min-max problems and algorithms by a relaxation procedure," *IEEE Transactions on Automatic Control*, vol. 25, pp. 62–66, feb 1980.

[42] A. Zhou and Q. Zhang, "A surrogate-assisted evolutionary algorithm for minimax optimization," *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, pp. 0–6, 2010.

[43] J. Marzat, E. Walter, and H. Piet-Lahanier, "Worst-case global optimization of black-box functions through Kriging and relaxation," *Journal of Global Optimization*, vol. 55, no. 4, pp. 707–727, 2013.

[44] E. Zeeman, "Catastrophe theory.," *Scientific American*, 1976.

[45] O. Makarenkov and J. S. Lamb, "Dynamics and bifurcations of nonsmooth systems: A survey," *Physica D: Nonlinear Phenomena*, vol. 241, no. 22, pp. 1826–1844, 2012.

[46] G. J. Klir, "Is there more to uncertainty than some probability theorists might have us believe?," *International Journal of General Systems*, vol. 15, no. 4, pp. 347–378, 1989.

[47] George, J., Klir·, Mark, and Wiennan, "Uncertainty - Based Information :Elements of Generalized Information Theory," in *Nato science series sub series III computer and systems sciences*, vol. 184, pp. 21–52, 2003.

[48] G. J. Klir, "Generalized information theory: Aims, results, and open problems," in *Reliability Engineering and System Safety*, vol. 85, pp. 21–38, 2004.

[49] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. J. Sallaberry, "Representation of analysis results involving aleatory and epistemic uncertainty," *International Journal of General Systems*, vol. 39, no. 6, pp. 605–646, 2010.

[50] A. P. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *https://doi.org/10.1214/aoms/1177698950*, vol. 38, pp. 325–339, apr 1967.

[51] G. Shafer, "A mathematical theory of evidence," p. 297, 1976.

[52] A. V. Allan Barrat, Marc Barthlemy, "Dynamical Processes on Complex Networks," in *Scrooge Meets Dick and Jane*, pp. 28–28, 2021.

[53] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, "Complex networks: Structure and dynamics," feb 2006.

[54] S. Division, "Space engineering. Engineering design model data exchange (CDF)," Tech. Rep. ECSS-TM-E-10-25A 20 October 2010 First, 2010.

[55] G. Filippi and M. Vasile, "Global Solution of Constrained Min-Max Problems with Inflationary Differential Evolution," in *Optimisation in Space Engineering OSE* (E. Minisci, A. Riccardi, and M. Vasile, eds.), no. Optimization and Engineering, Springer, 2020.

[56] G. Filippi and M. Vasile, "A Multi Layer Evidence Network Model for the Design Process of Space Systems Under Epistemic Uncertainty," in *Computational Methods in Applied Sciences*, vol. 55, pp. 227–243, Springer Science and Business Media B.V., 2021.

[57] G. Filippi and M. Vasile, "Network Resilience Optimisation of Complex Systems," pp. 315–326, 2021.

[58] G. Filippi and M. Vasile, "Introduction to Evidence-Based Robust Optimisation," in *Optimization Under Uncertainty with Applications to Aerospace Engineering*, pp. 541–573, Springer, Cham, 2021.

[59] G. Filippi and M. Vasile, "Inflationary Differential Evolution for Constrained Multi-objective Optimisation Problems," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12438 LNCS, pp. 29–42, Springer, Cham, nov 2020.

[60] C. O. Absil and M. Vasile, "A Variance-Based Estimation of the Resilience Indices in the Preliminary Design Optimisation of Engineering Systems Under Epistemic Uncertainty," in *EUROGEN*, 2017.

[61] M. Vasile, G. Filippi, C. Ortega Absil, and A. Riccardi, "Fast belief estimation in evidence network models," in *EUROGEN*, (Madrid), sep 2017.

[62] G. Filippi, M. Marchi, M. Vasile, and P. Vercesi, "Evidence-based robust optimisation of space systems with evidence network models," 2018.

[63] G. Filippi, M. Vasile, P. Z. Korondi, M. Marchi, and C. Poloni, "Robust design optimisation of dynamical space systems," in *8th International Systems & Concurrent Engineering for Space Applications Conference*, (Glasgow), 2018.

[64] G. Filippi, D. Krpelik, P. Z. Korondi, M. Vasile, M. Marchi, and C. Poloni, "Space systems resilience engineering and global system reliability optimisation under imprecision and epistemic uncertainty," in *Proceedings of the International Astronautical Congress, IAC*, vol. 2018-Octob, (Bremen), 2018.

[65] G. Filippi and M. Vasile, "Evidence-based resilience engineering of dynamic space systems," in *Proceedings of the International Astronautical Congress, IAC*, vol. 2019-Octob, (Washington), 2019.

[66] G. Filippi and M. Vasile, "A Multi Layer Evidence Network Model for the Design Process of Space Systems under Epistemic Uncertainty," in *EUROGEN*, (Guimaraes), 2019.

[67] G. Filippi and M. Vasile, "A Memetic Approach to the Solution of Constrained Min-Max Problems," in *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, (Wellington), pp. 506–513, 2019.

[68] C. Greco, L. Gentile, G. Filippi, E. Minisci, M. Vasile, and T. Bartz-Beielstein, "Autonomous Generation of Observation Schedules for Tracking Satellites with Structured-Chromosome GA Optimisation," in *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, (Wellington), pp. 497–505, 2019.

[69] G. Filippi, D. Gillespie, A. Ross Wilson, and M. Vasile, "A resilience approach to the design of future Moon base power systems," in *Int. Astronaut. Congr. IAC*, 2020.

[70] G. Filippi, M. Vasile, E. Patelli, and M. Fossati, "Generative Optimisation of Resilient Drone Logistic Networks," in *2022 IEEE Congress on Evolutionary Computation, CEC 2022 - Conference Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022.

[71] G. Filippi, M. Vasile, E. Patelli, and M. Fossati, "Resilience Optimisation for Next Generation Drone Logistic Networks," in *European Conference on Safety and Reliability (ESREL)*, 2022.

[72] D. Gillespie, A. R. Wilson, D. Martin, G. Mitchell, G. Filippi, and M. Vasile, "Comparative analysis of solar power satellite systems to support a moon base," in *Proceedings of the International Astronautical Congress, IAC*, vol. 2020-Octob, 2020.

[73] L. Gentile, G. Filippi, E. Minisci, M. Vasile, T. Bartz-Beielstein, and M. Vasile, "Preliminary spacecraft design by means of Structured-Chromosome Genetic Algorithms," in *IEEE Congress on Evolutionary Computation (CEC)*, (Glasgow), 2020.

[74] A. W. Wymore, *Model-Based Systems Engineering*. Boca Raton, Florida: C. Press, 1993.

[75] R. Levins, "Strategies of abstraction," *Biology and Philosophy*, vol. 21, no. 5, pp. 741–755, 2006.

[76] T. a. Zang, M. J. Hemsch, M. W. Hilburger, S. P. Kenny, J. M. Luckring, P. Maghami, S. L. Padula, and W. J. Stroud, "Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles," *NASA Technical Reports Server (NTRS)*, vol. 211462, no. July, p. paste, 2002.

[77] N. Croisard, M. Vasile, S. Kemble, and G. Radice, "Preliminary space mission design under uncertainty," *Acta Astronautica*, vol. 66, pp. 654–664, mar 2010.

[78] E. E. Lewis, *Introduction to Reliability Engineering*. John Wiley and Sons, Inc, 1994.

[79] W. Yao, X. Chen, W. Luo, M. van Tooren, and J. Guo, "Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles," *Progress in Aerospace Sciences*, vol. 47, pp. 450–479, aug 2011.

[80] D. Woods, "Creating foresight: How resilience engineering can transform NASA's approach to risky decision making," *Work*, vol. 4, no. 2, pp. 137–144, 2003.

[81] A. M. Madni and S. Jackson, "Towards a conceptual framework for resilience engineering," *IEEE Systems Journal*, vol. 3, pp. 181–191, jun 2009.

[82] S. A. Sheard, "Twelve Systems Engineering Roles," *INCOSE International Symposium*, vol. 6, no. 1, pp. 478–485, 2014.

[83] NASA NPR7120.5, "NPR 7120 . 5, NASA Space Flight Program and Project Management Handbook," *NASA's Procedural Requirements*, no. February, 2010.

[84] S. Hirshorn, "NASA System Engineering Handbook SP-2016-6105 Rev2," p. 297, 2016.

[85] ESA, "Space project management - project phasing and planning," in *ECSS Secretariat ESA-ESTEC Requirements & Standards Division*, 1996.

[86] N. N. Taleb, *The Black Swan*. 2009.

[87] V. Jacobson, "Congestion avoidance and control," in *Symposium Proceedings on Communications Architectures and Protocols, SIGCOMM 1988*, pp. 314–329, 1988.

[88] R. Guimerà, A. Arenas, A. Díaz-Guilera, and F. Giralt, "Dynamical properties of model communication networks," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 66, no. 2, 2002.

[89] M. L. Sachtjen, B. A. Carreras, and V. E. Lynch, "Disturbances in a power transmission system," *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, vol. 61, no. 5, pp. 4877–4882, 2000.

[90] J. Wang, B. Xu, and Y. Wu, "Ability paradox of cascading model based on betweenness," *Scientific Reports*, vol. 5, 2015.

[91] S. Dunn and S. M. Wilkinson, "Increasing the resilience of air traffic networks using a network graph theory approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 90, pp. 39–50, jun 2016.

[92] S. Dunn, G. Fu, S. Wilkinson, and R. Dawson, "Network theory for infrastructure systems modelling," in *Proceedings of the Institution of Civil Engineers: Engineering Sustainability*, vol. 166, pp. 281–292, oct 2013.

[93] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. De Porcellinis, and R. Setola, "Modelling interdependent infrastructures using interacting dynamical models," *International Journal of Critical Infrastructures*, vol. 4, no. 1-2, pp. 63–79, 2008.

[94] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[95] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[96] G. A. K. Van Voorn, "PhD mini course: introduction to bifurcation analysis," tech. rep.

[97] S. Sharma, E. B. Coetzee, M. H. Lowenberg, S. A. Neild, and B. Krauskopf, "Numerical continuation and bifurcation analysis in aircraft design: an industrial perspective,"

[98] A. J. Homburg and B. Sandstede, "Homoclinic and heteroclinic bifurcations in vector fields," *Handbook of Dynamical Systems*, vol. 3, no. C, pp. 379–524, 2010.

[99] M. M. Bosschaert, S. G. Janssens, and Y. A. Kuznetsov, "Switching to nonhyperbolic cycles from codimension two bifurcations of equilibria of delay differential equations," mar 2019.

[100] Q. Li, X. S. Yang, and S. Chen, "Hyperchaos in a spacecraft power system," *International Journal of Bifurcation and Chaos*, vol. 21, no. 6, pp. 1719–1726, 2011.

[101] A. Borshchev and A. Filippov, "From System Dynamics and Discrete Event to Practical Agent Based Modeling : Reasons , Techniques , Tools 1 . Simulation Modeling : Abstraction Levels , Major Paradigms," *22nd International Conference of the System Dynamics Society, 25-29 July 2004*, p. 45, 2004.

[102] J. Kofránek, M. Mateják, P. Privitzer, and M. Tribula, "CAUSAL OR ACAUSAL MODELLING: LABOUR FOR HUMANS OR LABOUR FOR MACHINES,"

[103] A. L. Jousselme, P. Maupin, and É. Bossé, "Uncertainty in a situation analysis perspective," *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*, vol. 2, pp. 1207–1213, 2003.

[104] G. J. Klir, "Where do we stand on measures of uncertainty, ambiguity, fuzziness, and the like?," *Fuzzy Sets and Systems*, vol. 24, no. 2, pp. 141–160, 1987.

[105] P. Smets, "Imperfect information : Imprecision-Uncertainty," 1999.

[106] F. W, *An introduction to probability theory and its applications*. New York: Wiley, 2 ed., 1971.

[107] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 100, pp. 9–34, 1999.

[108] J. Liu, J. Yang, J. Wang, and H. Sii, "Review of Uncertainty Reasoning Approaches as Guidance for Maritime and Offshore Safety-Based Assessment," *Safety and Reliability*, vol. 23, no. 1, pp. 63–80, 2002.

[109] S. H. Lee and W. Chen, "A comparative study of uncertainty propagation methods for black-box-type problems," 2009.

[110] H. Madsen, S. Krenk, and N. Lind, *Methods of Structural Safety*. Dover Publications, 2 ed., 2006.

[111] K. Cheng, Z. Lu, and Y. Zhen, "Multi-level multi-fidelity sparse polynomial chaos expansion based on gaussian process regression," *Computer Methods in Applied Mechanics and Engineering*, vol. 349, pp. 360–377, 2019.

[112] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. J. Sallaberry, "Sensitivity analysis in conjunction with evidence theory representations of epistemic uncertainty," *Reliability Engineering and System Safety*, vol. 91, no. 10-11, pp. 1414–1434, 2006.

[113] J. C. Helton, J. D. Johnson, W. L. Oberkampf, and C. B. Storlie, "A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, pp. 3980–3998, aug 2007.

[114] C. Joslyn and J. C. Helton, "Bounds on belief and plausibility of functionally propagated random sets," *Annual Conference of the North American Fuzzy Information Processing Society - NAFIPS*, vol. 2002-January, pp. 412–417, 2002.

[115] J. Cliff, V. Kreinovich, C. Joslyn, and , Vladik Kreinovich, "Convergence properties of an interval probabilistic approach to system reliability estimation," *Journal of Theoretical Biology*, 2007.

[116] M. Di Carlo, M. Vasile, C. Greco, and R. Epenoy, "Robust optimisation of low-thrust interplanetary transfers using evidence theory," in *Advances in the Astronautical Sciences*, vol. 168, pp. 339–358, 2019.

[117] F. Voorbraak, "A computationally efficient approximation of Dempster-Shafer theory," *International Journal of Man-Machine Studies*, vol. 30, pp. 525–536, may 1989.

[118] D. Dubois and H. Prade, "Consonant approximations of belief functions," *International Journal of Approximate Reasoning*, vol. 4, pp. 419–449, 1990.

[119] B. Tessem, "Approximations for efficient computation in the theory of evidence," *Artificial Intelligence*, vol. 61, no. 2, pp. 315–329, 1993.

[120] M. Bauer, "Approximations for Decision Making in the Dempster-Shafer Theory of Evidence," in *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pp. 73–80, 1996.

[121] D. Han, J. Dezert, and Y. Yang, "Two novel methods for BBA approximation based on focal element redundancy," in *2015 18th International Conference on Information Fusion, Fusion 2015*, pp. 428–434, 2015.

[122] T. Denœux, "Inner and Outer Clustering Approximations of Belief Structures," 2000.

[123] D. Harmanec, "Faithful approximations of belief functions," *Proc. 15th Conf. Uncertainty in artificial intelligence (UAI)*, pp. 271–278, 1999.

[124] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[125] M. Schmidt and Y. Beck, "A gentle and incomplete introduction to bilevel optimization," 2021.

[126] R. Dawkins, *The Selfish Gene*. 2006.

[127] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution*. Springer-Verlag, 2005.

[128] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 4–31, 2 2011.

[129] M. Vasile, E. Minisci, and M. Locatelli, "An inflationary differential evolution algorithm for space trajectory optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 267–281, apr 2011.

[130] M. Locatelli and M. Vasile, "(non) convergence results for the differential evolution method," *Optimization Letters*, vol. 9, no. 3, pp. 413–425, 2015.

[131] M. Di Carlo, M. Vasile, and E. Minisci, "Adaptive multi-population inflationary differential evolution," *Soft Computing*, vol. 24, no. 5, pp. 3861–3891, 2020.

[132] D. J. Wales and J. P. Doye, "Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms," *Journal of Physical Chemistry A*, vol. 101, no. 28, pp. 5111–5116, 1997.

[133] R. W. Chaney, "A method of centers algorithm for certain minimax problems," *Mathematical Programming*, vol. 22, pp. 202–226, dec 1982.

[134] H. Aissi, C. Bazgan, and D. Vanderpooten, "Min-max and min-max regret versions of combinatorial optimization problems: A survey," *European Journal of Operational Research*, vol. 197, no. 2, pp. 427–438, 2009.

[135] D. Indraneel, "Robustness optimization for constrained nonlinear programming problems," *http://dx.doi.org/10.1080/03052150008941314*, vol. 32, pp. 585–618, 2007.

[136] J. M. Mulvey and S. A. Zenios, "Robust optimization of large-scale systems," *Operations Research*, vol. 43, pp. 264–281, 1995.

[137] W. Chen, M. M. Wiecek, and J. Zhang, "Quality utilitya compromise programming approach to robust design," *Journal of Mechanical Design*, vol. 121, pp. 179–187, 6 1999.

[138] N. Rolander, J. Rambo, Y. Joshi, J. K. Allen, and F. Mistree, "An approach to robust design of turbulent convective systems," *Journal of Mechanical Design*, vol. 128, pp. 844–855, 7 2006.

[139] Y. Jin and B. Sendhoff, "Trade-off between performance and robustness: An evolutionary multiobjective approach," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2632, pp. 237–251, 2003.

[140] W. Oberkampf and J. Helton, "Investigation of evidence theory for engineering applications," 4 2002.

[141] W. Yao, X. Chen, Y. Huang, Z. Gurdal, and M. Van Tooren, "Sequential Optimization and Mixed Uncertainty Analysis Method for Reliability-Based Optimization," *AIAA Journal*, vol. 51, pp. 2266–2277, sep 2013.

[142] X. Hu, X. Chen, V. Lattarulo, and G. T. Parks, "Multidisciplinary optimization under high-dimensional uncertainty for small satellite system design," *AIAA Journal*, vol. 54, 2016.

[143] J. C. Helton, *Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty*, vol. 57. 1997.

[144] Z. L. Huang, C. Jiang, Z. Zhang, T. Fang, and X. Han, "A decoupling approach for evidence-theory-based reliability design optimization," *Structural and Multidisciplinary Optimization*, vol. 56, pp. 647–661, sep 2017.

[145] Z. P. Mourelatos and J. Zhou, "A design optimization method using evidence theory," *Journal of Mechanical Design, Transactions of the ASME*, vol. 128, no. 4, pp. 901–908, 2006.

[146] A. M. Abdelbar, S. Ragab, and S. Mitri, "Applying Co-Evolutionary Particle Swam Optimization to the Egyptian Board Game Seega," *Proceedings of the First Asian-Pacific Workshop on Genetic Programming*, pp. 9–15, 2003.

[147] E. J. Hughes, "Checkers using a co-evolutionary on-line evolutionary algorithm," *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings*, vol. 2, pp. 1899–1905, 2005.

[148] C. E. Shannon, "Philosophical Magazine Series 7 XXII. Programming a computer for playing chess," *Philosophical Magazine Series*, vol. 7314, no. 41, pp. 256–275, 1950.

[149] D. Agnew, "Improved Minimax Optimization for Circuit Design," aug 1981.

[150] A. V. Sebald and J. Schlenzig, "Minimax Design of Neural Net Controllers for Highly Uncertain Plants," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 73–82, 1994.

[151] Y. S. Ong, P. B. Nair, and K. Y. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 392–404, 2006.

[152] T. M. Cavalier, W. A. Conner, E. del Castillo, and S. I. Brown, "A heuristic algorithm for minimax sensor location in the plane," *European Journal of Operational Research*, vol. 183, no. 1, pp. 42–55, 2007.

[153] J. Kim and M. J. Tahk, "Co-evolutionary computation for constrained min-max problems and its applications for pursuit-evasion game," *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, vol. 2, pp. 1205–1212, 2001.

[154] A.S Lewis, "Robust Regularization," tech. rep., 2002.

[155] J. W. Herrmann, "A genetic algorithm for minimax optimization problems," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 2, pp. 1099–1103, 1999.

[156] L. E. Ghaoui, H. Herv, and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *Society for Industrial and Applied Mathematics*, vol. 18, p. 15, 1997.

[157] A. M. Cramer, S. D. Sudhoff, and E. L. Zivi, "Evolutionary algorithms for minimax problems in robust design," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 444–453, apr 2009.

[158] R. I. Lung and D. Dumitrescu, "A new evolutionary approach to minimax problems," *2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 1902–1905, 2011.

[159] T. M. Cavalier, W. A. Conner, E. del Castillo, and S. I. Brown, "A heuristic algorithm for minimax sensor location in the plane," *European Journal of Operational Research*, vol. 183, no. 1, pp. 42–55, 2007.

[160] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for minimax problems," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, pp. 1576–1581, 2002.

[161] M. J. Tahk and B. C. Sun, "Coevolutionary augmented lagrangian methods for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 114–124, 2000.

[162] Y. Shi and R. A. Krohling, "Co-evolutionary particle swarm optimization to solve min-max problems," *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, vol. 2, pp. 1682–1687, 2002.

[163] M. T. Jensen, "A New Look at Solving Minimax Problems with Coevolution," in *4th Metaheuristics International Conference*, 2001.

[164] G. J. O. Tamer Basar, *Dynaminc Noncooperative Game Theory*. Academic Press, 1982.

[165] H. J. Barbosa, "A coevolutionary genetic algorithm for constrained optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*, vol. 3, pp. 1605–1611, 1999.

[166] M. T. Jensen, *Robust and Flexible Scheduling with Evolutionary Computation*. PhD thesis, 2001.

[167] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, *Lecture Notes in Computer Science*, vol. 5199 LNCS. 2008.

[168] B. Rustem, *Algorithms for Nonlinear Programming and Multiple Objective Decisions*. 1998.

[169] A. Sinha, Z. Lu, K. Deb, and ·. P. Malo, "Bilevel optimization based on iterative approximation of multiple mappings," *Journal of Heuristics*, vol. 26, pp. 151–185, 2020.

[170] A. Sinha, P. Malo, and K. Deb, "A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications," in *IEEE Transactions on Evolutionary Computation*, vol. 22, pp. 276–295, 2018.

[171] H. Agarwal, J. E. Renaud, E. L. Preston, and D. Padmanabhan, "Uncertainty quantification using evidence theory in multidisciplinary design optimization," in *Reliability Engineering and System Safety*, vol. 85, pp. 281–294, Elsevier, jul 2004.

[172] A. Gaiddon, J.-N. Greard, and D. Pagan, "Automated Optimization of Supersonic Missile Performances Taking into Account Design Uncertainties," in *33rd AIAA Fluid Dynamics Conference and Exhibit*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, jun 2003.

[173] K. M. Miettinen, *Nonlinear Multiobjective Optimization*. Dordrecht: Kluwer Academic Publishers Group, 2012.

[174] M. T. Emmerich and A. H. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods," *Natural Computing*, vol. 17, no. 3, pp. 585–609, 2018.

[175] K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms: An Introduction," in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing* (L. Springer, ed.), pp. 1–24, 2011.

[176] E. K. Burke and K. Graham, *Search methodologies: Introductory tutorials in optimization and decision support techniques*. Springer US, jan 2014.

[177] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Engineering*, vol. 5, no. 1, pp. 1–16, 2018.

[178] K. Deb, "Multi-objective genetic algorithms: problem difficulties and construction of test problems.," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1998.

[179] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results.," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[180] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1993, pp. 284–298, 2001.

[181] R. Kasimbeyli, Z. K. Ozturk, N. Kasimbeyli, G. D. Yalcin, and B. I. Erdem, "Comparison of some scalarization methods in multiobjective optimization: Comparison of Scalarization Methods," *Bulletin of the Malaysian Mathematical Sciences Society*, vol. 42, pp. 1875–1905, sep 2019.

[182] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer, "On a bicriterion formation of the problems of integrated system identification and system optimization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-1, no. 3, pp. 296–297, 1971.

[183] S. Gass and T. Saaty, "The computational algorithm for the parametric objective function," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 39–45, 1955.

[184] H. P. Benson, "Existence of efficient solutions for vector maximization problems," *Journal of Optimization Theory and Applications*, vol. 26, no. 4, pp. 569–580, 1978.

[185] V. J. Bowman, "On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives," in *Multiple Criteria Decision Making. Lecture Notes in Economics and Mathematical Systems (Operations Research)* (Z. S. Thiriez H., ed.), vol. 130, pp. 76–86, Berlin, Heidelberg: Springer-Verlag, 1976.

[186] R. I. Bo, S. M. Grad, and G. Wanka, "A general approach for studying duality in multiobjective optimization," *Mathematical Methods of Operations Research*, vol. 65, no. 3, pp. 417–444, 2007.

[187] G. Eichfelder, "Scalarizations for adaptively solving multi-objective optimization problems," *Computational Optimization and Applications*, vol. 44, no. 2, pp. 249–273, 2009.

[188] R. N. Gasimov, "Characterization of the Benson Proper Efficiency and Scalarization in Nonconvex Vector Optimization," in *Multiple Criteria Decision Making*

in the New Millennium (M. Köksalan, ed.), pp. 189–198, Berlin, Heidelberg: Springer-Verlag, 2001.

[189] M. Vasile, "On the solution of min-max problems in robust optimization," in *The EVOLVE 2014 International Conference, A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computing*, (Jian-Guo Hotel), Jian-Guo Hotel, 2014.

[190] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[191] R. Baxter, N. Hastings, A. Law, and E. J. Glass, *Algorithms for worst-case design and applications to risk management*, vol. 39. 2008.

[192] M. Jamil and X.-S. Yang, "A Literature Survey of Benchmark Functions For Global Optimization Problems," *Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.

[193] S. Alicino and M. Vasile, "An evolutionary approach to the solution of multi-objective min-max problems in evidence-based robust optimization," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, 2014.

[194] D. Vallado and W. McClain, *Fundamentals of Astrodynamics and Applications*. Fundamentals of Astrodynamics and Applications, Microcosm Press, 2001.

[195] M. Patel and O. Beik, *Spacecraft Power Systems*. CRC Press, 2023.

[196] J. R. Norris, *Markov Chains*. New York: Cambridge University Press, 2009.

[197] G. Fubini, "Sugli integrali multipli," in *Rend. Acc. Naz. Lincei*, pp. 608–614, 1907.

[198] G. Benettin, "Una passeggiata tra i Sistemi Dinamici," 2012.

[199] R. Seydel, "Basic Bifurcation Phenomena," *Computer*, vol. 49, no. June, 1999.

[200] B. Barzel and A. L. Barabási, "Universality in network dynamics," *Nature Physics*, vol. 9, no. 10, pp. 673–681, 2013.

[201] R. M. Sumanth, "Computation of eclipse time for low-earth orbiting small satellites," *International Journal of Aviation, Aeronautics and Aerospace*, 2019.

[202] G. Gordon and W. Morgan, *Principles of communication satellites*. John Wiley and sons, Inc., 1993.

[203] "Features Heritage:-3 units for world's fastest communication satellite "KIZUNA" (WINDS)-8 units for greenhouse gas observation technology satellite "IBUKI" (GOSAT)-3 units for Quasi-Zenith Satellite "MICHIBIKI" (QZSS)-6 units for Global Change Observation Mission 1st-Water SHIZUKU" (GCOM-W1)," tech. rep.

[204] "Features Heritage:-3 units for world's fastest communication satellite "KIZUNA" (WINDS)-8 units for greenhouse gas observation technology satellite "IBUKI" (GOSAT)-3 units for Quasi-Zenith Satellite "MICHIBIKI" (QZSS)-6 units for Global Change Observation Mission 1st-Water SHIZUKU" (GCOM-W1)," tech. rep.

[205] "ECAM-DVR4 Digital Video Recorder, 4-Port Features ECAM-DVR4 Application ECAM-DVR4 Malin Space Science Systems Exploration Through Imaging Space Cameras and Systems," tech. rep.

[206] C. M. Evan Clinton, Andris Jaunzemis and F. Wang, "Satellite downlink."

[207] VECTRONIC-Aerospace, "Payload data handling system vpdhs-vectronic aerospace [online]," 2024.

[208] C. Brown, *Elements of Spacecraft Design.* AIAA Education Series, 2002.

[209] J. Wertz and W. Larson, *Space Mission Analysis and Design.* 3rd ed., Microcosm Press, 1999.

[210] W. Ley, K. Wittmann, and W. Hallmann, *Handbook of Space Technology.* 2009.

[211] J. F. Castet and J. H. Saleh, "Satellite and satellite subsystems reliability: Statistical data analysis and modeling," *Reliability Engineering and System Safety*, vol. 94, no. 11, pp. 1718–1728, 2009.

[212] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 34, no. 2, pp. 187–220, 1972.

[213] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems," *Structural Optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[214] M. Vasile and E. Minisci, "Approximated computation of belief functions for robust design optimization," pp. 1–18, 2012.

[215] J. C. Helton, J. D. Johnson, and W. L. Oberkampf, "An exploration of alternative approaches to the representation of uncertainty in model predictions," in *Reliability Engineering and System Safety*, vol. 85, pp. 39–71, 2004.

[216] F. Zuiani and M. Vasile, "Multi Agent Collaborative Search based on Tchebycheff decomposition," *Computational Optimization and Applications*, vol. 56, pp. 189–208, sep 2013.

Bibliography