

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria Aerospaziale



Optimization under Uncertainty

theory, algorithms and industrial applications

Ph.D. Industrial Engineering

Supervisors:
Prof. Carlo de Nicola
Ing. Domenico Quagliarella

Candidate:
Giovanni Petrone

Reviewers:
Prof. Gianluca Iaccarino
Prof. Lina Mallozzi

January 2012

Abstract

Uncertainty quantification of numerical simulations has raised significant interest in recent years and, as a consequence, the interest in a procedure of optimization under uncertainty. One of the main challenges in this field is the efficiency in propagating uncertainties from the sources to the quantities of interest, especially when there are many sources of uncertainties. Other important challenges are the coupling of the optimization procedure with the uncertainty quantification routines, usually approached as two independent problems, and the necessity to perform efficiently a massive ensemble of numerical simulations.

The primary goals of this work are to develop algorithms for efficient uncertainty quantification and optimization under uncertainty and to use them in industrial applications. We first introduce the a novel way to perform uncertainty quantification based on simplex elements on the probability space and we prove its effectiveness in real life problems. We prove that this algorithm requires a fewer number of evaluations of the quantity of interest with respect to widely used approach adopted in this field of study. This is particular important in a process of optimization under uncertainty where the cost of the deterministic optimization is raised up by the presence of a

nested uncertainty quantification algorithm.

We will review the state of the art for optimization under uncertainty in order to introduce novel methodologies that overcome the limitations of the actual framework. These novel formulations contemplate the full identity card of a system analyzed under uncertainty - the Cumulative Distribution Function. A methodology to approach single-objective problems with an a posteriori selection of the candidate design based on risk/opportunity criteria of the designer will be presented and assessed. Therefore multi-objective problems will be considered and a novel algorithm will be presented, the *P-NSGA* (Probabilistic Non-dominated sorted Genetic Algorithm), that generalize the *NSGA-II*, a widely adopted algorithm for multi-objective deterministic optimization.

Furthermore the cost of optimization under uncertainty motivates the effort that will be given to High Performance Computing in order to obtain the most efficient solution to perform automatically a large ensemble of computations. We will present *Leland*, a simulation environment that has been developed to dynamically *schedule, monitor and stir* the calculation ensemble and extract runtime information as well as simulation results and statistics. *Leland* is equipped with an auto-tuning strategy for optimal load balancing and fault tolerance checks to avoid failures in the ensemble – features that will be proven to be a necessity in optimization under uncertainty.

Game Theory will be investigated and proven to be a possible solution in handling problems of optimization under uncertainty where a lack of knowledge about the variability of several uncertain parameters is taken in account.

Two industrial applications will be presented in the development of this thesis: the optimization of the shape of wind turbine blades and the optimization of a Formula 1 tire brake intake. Both problems are multi-objective and the presence of uncertainties significantly impact on the estimation of their responses, hence they are well-suited to assess the theoretical framework and the algorithms that will be presented in this thesis.

Acknowledgments

I want to first thank my advisor, Professor Carlo de Nicola. In every discussion, his broad knowledge and deep insight has helped develop the directions for my research.

I also thank my co-advisor, Ing. Domenico Quagliarella (Centro Italiano Ricerche Aerospaziali, CIRA). Throughout my Ph.D. study, he has provided helpful guidance in the field of Optimization, good company and many great ideas.

I've been lucky since I had two additional supervisors who gave a precious guidance for the arguments developed in this thesis.

Indeed I first gratefully acknowledge Professor Gianluca Iaccarino (Stanford University) for his efforts in training me on Uncertainty Quantification and for giving me the opportunity to spend a full year in his research group at the Center for Turbulence Research. He gave me a lot of opportunities, resources and support to develop my ideas.

Then I would like to thank Professor Lina Mallozzi who took care of training me on Game Theory and to introduce me to the wonderful field of mathematics and cultivate my interest in it.

I am deeply grateful to the Ph.D. student John Axerio-Cilies who worked with me on various projects. We worked hard together everyday, till night, during my research period at Stanford University. All the Formula 1 activities that have been field of application for my optimization ideas come from his Ph.D. research. I am either deeply grateful to the postdoctoral fellow Jeroen Witteveen who worked with me on the development of methodologies for uncertainty quantification.

Discussions with Professor Juan Alonso and Professor Karthik Duraisamy were also helpful. I am indebted to my many friends and colleagues in the Building 500 of Stanford University for providing such an excellent environment for learning and research. I am especially grateful to Mike Emory and Gary Tang for fruitful discussions. Additionally I am indebted to my many friends and colleagues in University Federico II of Naples. I am especially grateful to Daniele Elia and Egidio d'Amato for the work done on Algorithmic Game Theory.

Last, and most important, I wish to thank my family and my friends. This thesis would not be possible without their support. To them I dedicate my thesis.

List of Tables

1.1	AOC 15/50 wind turbine, principal characteristics	31
1.2	AOC 15/50 wind turbine, deterministic conditions	37
1.3	Analysis under uncertain meteorological conditions	41
1.4	Analysis under insect contamination	44
1.5	Analysis under manufacturing errors	46
2.1	Mass flow rate into the brake duct and drag force on the tire sensitivity for 9 uncertain variables and 3 design variables . . .	71
2.2	Multi-objective optimization strategy	73
2.3	Uncertainty quantification strategy	74
3.1	Single-objective optimization strategy	89
3.2	Relevant candidate designs	91
4.1	Multi-objective optimization strategy	116
6.1	The entropy-based game	149
6.2	GA details	159

List of Figures

1.1	Sampling based techniques with two uncertain inputs.(left) Monte Carlo; (center) Latin Hypercube; (right) Lattice based .	16
1.2	Legendre (left) and Hermite (right) polynomials	17
1.3	Simplex Stochastic Collocation discretization of a two-dimensional probability space.	21
1.4	NREL s827: $M=0.1$, $Re=3e6$, free transition	25
1.5	Breakdown of the noise generated by a 50kw wind turbine at a microphone located at $(x,y,z)=(-20m,0m,0m)$	28
1.6	EOLO flowchart	29
1.7	Wind speed, direction and turbulence intensity at the Acqua Spruzza, Italy site. The data is reported in terms of empirical probability distributions scaled from 40 to 24 meters.	33
1.8	Monte Carlo samples for meteorological conditions.	38
1.9	Meteorological conditions: cumulative density function of the power coefficient and Sound Pressure Level. The red lines represent the deterministic conditions.	39

1.10	Meteorological conditions: convergence histories of the mean, variance and error of the power coefficient	40
1.11	Meteorological conditions: convergence histories of the mean, variance and error of the Sound Pressure Level	40
1.12	Monte Carlo samples for insect contamination	42
1.13	Insect contamination: cumulative density function of the power coefficient and Sound Pressure Level	42
1.14	Insect contamination: convergence histories of the mean, variance and error of the power coefficient	43
1.15	Insect contamination: convergence histories of the mean, variance and error of the Sound Pressure Level	43
1.16	Monte Carlo samples for manufacturing errors	45
1.17	Manufacturing errors: cumulative density function of the power coefficient and Sound Pressure Level	45
1.18	Manufacturing errors: convergence histories of the mean, variance and error of the power coefficient	46
1.19	Manufacturing errors: convergence histories of the mean, variance and error of the Sound Pressure Level	46
1.20	Meteorological conditions: comparison of the mean and variance of the power coefficient for two different SSC and two different LHS	47
1.21	Meteorological conditions: comparison of the mean and variance of the Sound Pressure Level for two different SSC and two different LHS	47

2.1 Pareto Front and non-domination illustration 57

2.2 Wake sensitivity (shown by streamwise x-velocity contours for a plane located 1.12 wheel diameters downstream from the center of the tire) for a simplified tire with wheel fairings (top left), baseline F1 tire (top right), baseline F1 tire with blocked hub passages (bottom left), and simplified tire with artificial mass efflux from blue segment (bottom right) 62

2.3 Turbulent kinetic energy contours for the minimum drag configuration (top) and maximum cooling configuration (bottom) 63

2.4 Front right tire of the Formula 1 race car used in this study showing green airfoil strut used to secure tire to the experimental wind tunnel facility and the outer brake duct (magenta) used to cool the brake assembly 67

2.5 Four different views showing the Formula 1 tire mesh 68

2.6 Brake duct optimization variables 68

2.7 Subset of uncertain variables tested for sensitivity in output quantities of interest 69

2.8 Deterministic Pareto front (left); the green, blue, magenta, and gray brake ducts in the subfigure on the right correspond to the trade-off, max cooling, minimum drag, and baseline configurations respectively 72

2.9	Pareto frontier for F1 wheel assembly showing the variability of the minimum drag (magenta), baseline (red), trade-off (green), and maximum cooling (blue) designs to uncertainty in the inflow conditions and flexible tire geometry.	75
2.10	PDF's of the output quantities of interest used for this study .	75
3.1	Literature review of Optimization under Uncertainty	81
3.2	Probabilistic distance from a Uniform template	93
3.3	CDF based measure of robustness	94
3.4	Additive propriety of the RI	95
3.5	The Rosenbrock test function	96
3.6	The Goldstein-Price test function	96
3.7	Rosenbrock function. Pareto Front for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$	97
3.8	Rosenbrock function. CDFs for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$	98
3.9	Goldstein-Price function. Pareto Front for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$ 99	
3.10	Goldstein-Price function. CDFs for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$.	100
4.1	The use of Robustness Index as probabilistic rank	108
4.2	Non dominated solutions obtained with P-NSGA in comparison with deterministic Pareto.	115
4.3	Non dominated solutions obtained with P-NSGA in comparison with Mean Value Pareto and Mean Value Penalty Pareto.	117
4.4	CDFs of the objectives for the solution closest to probabilistic Pareto dominance.	118

4.5	Deterministic Pareto front	124
4.6	Probabilistic Pareto front	125
4.7	Probabilistic and deterministic trade-off designs	126
4.8	Chord and twist distributions	127
4.9	Wind turbine cross-sections	127
5.1	Leland flowchart	131
5.2	Sample HPC simulation diagnostics	133
5.3	Sample input file for Uncertainty Quantification	135
5.4	Web interface for dynamic inputs	135
5.5	The job liaison	136
5.6	Web interface for objective functions	137
5.7	Sample input file for Optimization under Uncertainty	138
6.1	Probability distribution function of the input uncertainty	148
6.2	Probability distribution function of the output	148
6.3	Average and best variables value through generation	158
6.4	Non dimensional time ratio through generation	159
6.5	Follower's objective function towards generations with multi-modal approach	167
6.6	Follower's objective function towards generations with an intermediate filter operation on multi-modal approach	168
6.7	Follower's objective function towards generations after filtering	168
6.8	Leader's objective function	169
6.9	Nash equilibrium GA structure	170

LIST OF FIGURES

xii

6.10 Stackelberg-Nash equilibrium GA structure	171
6.11 Stackelberg-GAs algorithm structure with non-uniqueness . . .	172

Contents

Abstract	i
Acknowledgments	iv
List of Tables	vi
List of Figures	vii
Introduction	1
1 Uncertainty Quantification	5
1.1 Definitions and basic concepts	7
1.2 Errors vs. uncertainties	7
1.2.1 Aleatory uncertainty	8
1.2.2 Epistemic uncertainty	9
1.2.3 Sensitivity vs. uncertainty analysis	9
1.3 Analysis under Uncertainty	10
1.3.1 Data assimilation	11
1.3.2 Uncertainty propagation	12
1.3.3 Certification	12

1.4	Non-intrusive propagation methods	13
1.4.1	Sampling techniques	14
1.4.2	Quadrature methods	16
1.4.3	The other approaches	18
1.5	Simplex Stochastic Collocation (SSC)	19
1.6	Case study: Analysis under Uncertainty of a Wind Turbine . .	22
1.6.1	EOL0: a multi-physics low-order model for wind turbines	23
1.6.2	The AOC 15/50	30
1.6.3	Analysis under Uncertainty	30
1.7	Lessons learned	48
2	Optimization and Uncertainty	49
2.1	The mathematical formulation	51
2.1.1	Design variables	52
2.1.2	Constraints	52
2.1.3	Objectives	52
2.1.4	Models	53
2.1.5	Standard form	53
2.2	Problem solution methods	54
2.2.1	Genetic algorithms	55
2.3	The Pareto front	56
2.4	The NSGA-II algorithm	57
2.5	Case study: Optimization of a F1 wheel assembly	60
2.5.1	The physics	64
2.5.2	The cost	65

2.5.3	The objectives	66
2.5.4	Optimization Variables	66
2.5.5	Uncertain Variables	67
2.5.6	Deterministic Pareto Front	70
2.5.7	Analysis under Uncertainty of Pareto Front	73
2.6	Lessons learned	76
3	Single-Objective Optimization under Uncertainty	77
3.1	The mathematical formulation	78
3.2	A novel CDF based approach	82
3.3	Global and local robustness	84
3.4	A CDF partition based approach	86
3.5	Single-objective analytic test functions	87
3.5.1	Rosenbrock function	88
3.5.2	Goldstein-Price function	90
3.6	Lessons learned	91
4	Multi-Objective Optimization under Uncertainty	101
4.1	The mathematical formulation	101
4.2	The generalization of the NSGA-II sorting operators	105
4.2.1	A probabilistic definition of the rank	106
4.2.2	A probabilistic definition of the crowding distance	107
4.2.3	A probabilistic Crowded-Comparison Operator	109
4.3	The algorithm	112
4.4	Multi-objective analytic test function	114

4.5	Case study: Multi-Objective Optimization under Uncertainty of a Wind Turbine	117
4.5.1	Definition of the shape optimization problem	118
4.5.2	Deterministic optimization	120
4.5.3	Optimization under Uncertainty	120
4.6	Lessons learned	122
5	Extreme Ensemble Computations	128
5.1	The need of High Performance Computing	128
5.2	Leland	130
5.3	The input file	134
5.4	The future of Leland	139
6	A bridge with games	140
6.1	Game Theory	141
6.1.1	Definition of game	141
6.1.2	The Nash equilibrium	142
6.1.3	The Stackelberg equilibrium	142
6.2	A novel entropy-based model	144
6.2.1	The maximum entropy principle	145
6.2.2	Entropy of a probability distribution	145
6.2.3	Optimization under Uncertainty	146
6.2.4	The follower	146
6.2.5	The leader	147
6.2.6	The game	149

6.3	The Stackelberg-Nash model	149
6.4	Algorithmic Game Theory	151
6.4.1	GA for a Nash equilibrium problem	152
6.4.2	GA for a Stackelberg-Nash equilibrium problem	155
6.4.3	Algorithm validation	156
6.4.4	Applications to real life problems	159
6.5	The non-uniqueness	160
6.5.1	Multi-modal optimization	162
6.5.2	Algorithm modifications for the non-uniqueness	163
6.5.3	A first numerical example	165
6.5.4	A second numerical example	166
7	Conclusions	173

Introduction

Planning under uncertainty. This, I feel, is the real field we should all be working on.

— G. B. Dantzig

Beginning with the seminal works of Beale [1], Bellman [2], Bellman and Zadeh [3], Charnes and Cooper [4], Dantzig [5], and Tintner [6], analysis, decision making and optimization under uncertainty have experienced rapid development in both theory and algorithms. Today, Dantzig still considers planning under uncertainty as one of the most important open problems in optimization (Horner [7]).

This thesis begins with the research of efficient methods for uncertainty quantification, then focuses on introducing them in the optimization procedure through novel approaches. Therefore Game Theory is used to define a process of optimization under uncertainty as a conflict between player when a lack of knowledge about the system is present. In industrial applications the response of a system is usually represented by complex multi-disciplinary simulations. Two industrial applications will be presented in the development

of this thesis to support the theoretical framework: the optimization under uncertainty of wind turbine blades and the optimization under uncertainty of a Formula 1 brake intake. High Performance Computing will be proven to be an important aspect in optimization under uncertainty due to the cost of these procedures.

In chapter 1, after introducing the basic concepts of uncertainty quantification, a novel methodology – the Simplex Stochastic Collocation – will be proven to be a suitable technique to be coupled with an optimization process. As an example of real life applications, this methodology will be used to assess the impact of different sources of uncertainty (e.g. uncertain meteorological conditions, insect contamination, manufacturing errors) on the performance and noise of a wind turbine, in order to prove the importance to take in account the variability of these conditions on the analysis of complex multi-disciplinary systems.

In chapter 2, after introducing the basic concepts of deterministic optimization, it will be shown the necessity of take in account uncertainty since the beginning of the design process. As an example of real life applications, the design of a F1 tire brake intake to maximize cooling efficiency and minimize aerodynamic resistance will be considered. It will be shown how the use of sophisticated and expensive techniques, such as three-dimensional Reynolds-Averaged Navier-Stokes simulations on a high performance computing cluster, could result meaningless due to the presence of uncertainties.

In chapter 3 the current framework for optimization under uncertainty is reviewed and a new methodology introduced. This novel formulation does

not rely on few statistical informations about the designs, but contemplate the full identity card of a system analyzed under uncertainty - the Cumulative Distribution Function. A metric will be introduced in order to guide a process of single-objective optimization under uncertainty, leading to an innovative approach for the a posteriori selection of the candidate design based on risk/opportunity criteria of the designer.

In chapter 4 the previously introduced methodologies will be extended to multi-objective optimization. Furthermore a novel probabilistic algorithm, the P-NSGA (Probabilistic Non-dominated sorted Genetic Algorithm), will be presented as generalization of the NSGA-II, a widely adopted algorithm for multi-objective deterministic optimization. The uncertainty quantification of wind turbine performance proposed in chapter 1 will be now embedded in an optimization process used to design blades less sensitive to environmental changes.

The industrial applications considered in this thesis clearly show that the computational resources needed to consider the uncertainty quantification in an optimization procedure require at least an increase in an order of magnitude in the number of evaluations of the response of the system with respect to deterministic optimization. Therefore it appears evident the need to perform extreme ensemble calculation in the most efficient way. This motivate the effort spent on High Performance Computing (HPC) in chapter 5. An HPC environment called *Leland* will be presented in details together with the features that have made it a well suited meta-scheduler for uncertainty quantification and optimization under uncertainty. *Leland* aims to a com-

plete fault tolerance in performing an ensemble of computation on clusters of CPUs.

The last chapter of this thesis focuses on Game Theory. The process of optimization under uncertainty will be seen as a conflict between players. This led to a new approach that could be particularly useful in presence of a lack of knowledge about the system. The classic concepts of Nash and Stackelberg equilibria are formulated in terms of entropy, a concept that comes from information theory and measures the content of informations of a probabilistic distribution. At the end of this chapter several generalizations will be proposed to handle multiple uncertainties and the non-uniqueness of the solution.

Chapter 1

Uncertainty Quantification

If a man will begin with certainties, he shall end in doubts; but if he will be content to begin with doubts he shall end in certainties.

— Francis Bacon

Several theories address the definition of uncertainty. These theories include probability theory (Green et al.[8]) (Helton et Oberkampf [9]), fuzzy set theory (Zimmermann [10]) and evidence theory (Bae et al. [11]) (Mourelatos et Zhou [12]). In this thesis, we work under the framework of probability theory, which provides a solid and comprehensive theoretical foundation and offers the most versatile statistical tools. In contrast to the traditional, deterministic simulations, we describe uncertainties as randomness, and model the sources of uncertainties as random variables, random processes and random fields. To quantify the sources of uncertainties, we must specify the joint probability density function of all these random variables, processes and fields. This step is usually very problem-dependent. The methods in-

volved in this step include statistical analysis, experimental error analysis and often expert judgment (Ellison et al. [13]). Although how to quantify model uncertainties and numerical uncertainties is still a topic of current research (Wojtkiewicz et al. [14]) (Draper [15]), successful examples exist of quantifying the uncertainty sources for very complex engineering systems. For example, Bose et Wright [16] were very successful in quantifying the uncertainties in the Martian atmosphere entry of the NASA Phoenix spacecraft. Once the sources of uncertainties are quantified, we need to calculate how these uncertainties propagate through the simulation to the quantities of interest. These, also known as objective functions, are the main quantities to be predicted. They are functions of all the random variables that describe the sources of uncertainty.

The final product of the uncertainty qualification process is a quantitative description of the likelihood in the values of the quantities of interest. It can only be obtained by combining our knowledge of the sources of uncertainties and the behavior of the objective functions with respect to these sources. In the probability theoretic framework, this quantitative description is a joint probability density function of the objective functions. The support of this joint probability density function, i.e., the space where the function is positive, describes all possible scenarios predicted by the computational simulation; in addition, the value of the probability density function indicates how likely each scenario is. This joint probability density function enables decision making based on risk analysis, removing the important limitations of deterministic computational simulations.

This chapter starts by defining precisely the basic concepts related to uncertainties; then the different phases of uncertainty quantification are introduced. Different methodologies will be presented in Section 1.4 but the focus of this thesis will be given to stochastic collocation in a novel implementation characterized by the use of simplex elements in the probability space, as shown in Section 1.5. An application to wind turbine analysis under uncertainty will be presented at the end of this chapter to prove the effectiveness of these novel methodologies in a complex multi-disciplinary framework. In chapter 4 we will define mathematical instruments to embed the presented process of uncertainty quantification in an optimization procedure, defining a novel way to design wind turbine blades in presence of uncertainties.

1.1 Definitions and basic concepts

The uncertainty quantification community has introduced precise definitions to characterize various types of uncertainties.

1.2 Errors vs. uncertainties

The American Institute of Aeronautics and Astronautics (AIAA) *Guide for the Verification and Validation of CFD Simulations* defines errors as recognizable deficiencies of the models or the algorithms employed and uncertainties as a potential deficiency that is due to lack of knowledge. This definition is not completely satisfactory because does not precisely distinguish between the mathematics and the physics. It is more useful to define errors as associated to the translation of a mathematical formulation into a

numerical algorithm (and a computational code). Errors are typically also further classified in two categories: acknowledged errors are known to be present but their effect on the results is deemed negligible. Examples are round-off errors and limited convergence of certain iterative algorithms. On the other end, unacknowledged errors are not recognizable [17] but might be present; implementation mistakes (bugs) or usage errors can only be characterized by comprehensive verification tests and procedures. Using the present definition of errors, the uncertainties are naturally associated to the choice of the physical models and to the specification of the input parameters required for performing the analysis. As an example, numerical simulations require the precise specification of boundary conditions and typically only limited information are available from corresponding experiments and observations. Therefore variability, vagueness, ambiguity and confusion are all factors that introduce uncertainties in the simulations. A more precise characterization is based on the distinction in aleatory and epistemic uncertainties.

1.2.1 Aleatory uncertainty

Aleatory uncertainty [18] is the physical variability present in the system being analyzed or its environment. It is not strictly due to a lack of knowledge and cannot be reduced. The determination of material properties or operating conditions of a physical system typically leads to aleatory uncertainties; additional experimental characterization might provide more conclusive evidence of the variability but cannot eliminate it completely. Aleatory uncertainty is normally characterized using probabilistic approaches.

1.2.2 Epistemic uncertainty

Epistemic uncertainty[19] is what is indicated in the AIAA Guide (AIAA 1998) as *uncertainty* [19], i.e. a potential deficiency that is due to a lack of knowledge. It can arise from assumptions introduced in the derivation of the mathematical model used or simplifications related to the correlation or dependence between physical processes. It is obviously possible to reduce the epistemic uncertainty by using, for example, a combination of calibration, inference from experimental observations and improvement of the physical models. Epistemic uncertainty is not well characterized by probabilistic approaches because it might be difficult to infer any statistical information due to the nominal lack of knowledge. A variety of approaches have been introduced to provide a more suitable framework for these analysis. Typical examples of sources of epistemic uncertainties are turbulence modeling assumptions and surrogate chemical kinetics models.

1.2.3 Sensitivity vs. uncertainty analysis

Sensitivity analysis (SA) investigates the connection between inputs and outputs of a (computational) model; more specifically, it allows to identify how the variability in an output quantity of interest is connected to an input in the model and which input sources will dominate the response of the system. On the other hand, uncertainty analysis aims at identifying the overall output uncertainty in a given system. The main difference is that sensitivity analysis does not require input data uncertainty characterization from a real device; it can be conducted purely based on the mathematical

form of the model. As a conclusion large output sensitivities (identified using SA) do not necessarily translate in important uncertainties because the input uncertainty might be very small in a device of interest. SA is often based on the concept of sensitivity derivatives, the gradient of the output of interest with respect to input variables. The overall sensitivity is then evaluated using a Taylor series expansion, which, to first order, would be equivalent to a linear relationship between inputs and outputs.

1.3 Analysis under Uncertainty

Computer simulations of an engineering device are performed following a sequence of steps. Initially the system of interest and desired performance measures are defined. The geometrical characterization of the device, its operating conditions, the physical processes involved are identified and their relative importance must be quantified. It is worthwhile to point out that the definition of the system response of interest is a fundamental aspect of this phase. The next step is the formulation of a mathematical representation of the system. The governing equations and the phenomenological models required to capture the relevant physical processes need to be defined. In addition, the precise geometrical definition of the device is introduced. This step introduces simplification with respect to the real system; for example small geometrical components are eliminated, or artificial boundaries are introduced to reduce the scope of the analysis. With a well defined mathematical representation of the system, the next step is to formulate a discretized representation. Numerical methods are devised to convert the continuous

form of the governing equations into an algorithm that produces the solution. This step typically requires, for example, the generation computational grid, which is a tessellation of the physical domain. Finally the analysis can be carried out. The introduction of uncertainty in numerical simulations does not alter this process but introduces considerable difficulties in each phase. It is useful to distinguish three steps: data assimilation, uncertainty propagation and certification.

1.3.1 Data assimilation

Data assimilation consists of a study of the system of interest that aims at identifying the properties, physical processes and other factors required to fully characterize it. The analysis is typically focused on the specific inputs required by the mathematical framework that will be applied in the simulations. As an example, the boundary conditions required in numerical simulations should be inferred from observation of the device of interest or specific experiments. Given the limited degree of reproducibility of experimental measurements and the errors associated to the measurement [20] techniques, these quantity are known with a certain degree of uncertainty. Probabilistic approaches treat these quantities, that overall characterize the aleatory uncertainty, as random variables assuming values within specified intervals. In mathematical terms this is equivalent to define random variables with a specified probability distribution functions (PDF). The obvious choice is to use random variables defined using analytical distributions (Gaussian, uniform, etc.). It is difficult to justify this choice [21] solely from experimen-

tal evidence because of the limited amount of data typically available⁶; in many situations the only data available are obtained from expert opinions and can lead to ambiguous or conflicting estimates. Alternative approaches have been devised to provide a more flexible framework to handle this situation, evidence theory is one such approach [22]. In the context of probabilistic approaches, the objective of data assimilation is to define PDFs of each of the input quantities used in the computational tool.

1.3.2 Uncertainty propagation

Once probability distributions are available for all the input quantities in the computational algorithm, the objective is to compute the PDFs of the output quantities of interest. This step is usually the most complex and computationally intensive for realistic engineering simulations. A variety of methods are available in the literature, from sampling based approaches (e.g. Monte Carlo) to more sophisticated stochastic spectral Galerkin approaches.

1.3.3 Certification

Once the statistics of the quantity of interest have been computed, various metrics can be used to characterize the system output, depending on the specific application. The most common use of such statistical information is a reliability assessment, where the likelihood of a certain outcome is estimated and compared to operating margins. In a validation context, the PDF (or more typically the cumulative distribution function) is compared to experimental observation to extract a measure of the confidence in the nu-

merical results. The characterization of these measures, so-called validation metrics, is an active area of research [23].

1.4 Non-intrusive propagation methods

Within a probabilistic framework, the problem of uncertainty propagation consists of the generation of PDFs of the outcomes given (known) distribution of all the input parameters. Several classes of methods have been developed to solve this problems; in this section three popular approaches are described. Consider the vector $\xi = [\xi_1, \dots, \xi_N]$ containing the input quantities to the computational model; assume that $f(\xi_i)$ is the output of interest; f is possibly the result of a complex fluid dynamic simulation. In probabilistic uncertainty quantification approaches the stochastic, input quantities x are represented as independent continuous random variables ξ mapping the sample space Ω_i to real numbers $\xi_i : \Omega_i \rightarrow R$. This assumption in practical terms increases the dimensionality of the problem: the original deterministic outcome $y = f(\xi_1, \dots, \xi_N)$ becomes a stochastic quantity. The objective is to compute the PDF of y in order to evaluate the likelihood of a certain outcome, or, in general, statistics of y . The expected value of y is defined as

$$\mu(y) = \int_{-\infty}^{\infty} y \Psi_y dy \quad (1.1)$$

where Ψ_y is the probability density function of y . Note that y is a stochastic variable while the expected value is deterministic quantities. Propagation method for uncertainty quantification can be classified in intrusive methods and non intrusive methods. A class of methods for uncertainty propagation

is based on spectral methods in which the solution is expressed using a suitable series expansion. These approaches are intrusive in the sense that the mathematical formulation requires modification of the existing deterministic codes. Despite these methodologies, in this thesis we focus on non-intrusive methods, namely the sampling approaches and the stochastic collocation methods.

1.4.1 Sampling techniques

Sampling-based techniques are the simplest approaches to propagate uncertainty in numerical simulations: they involve repeated simulations (also called realizations) with a proper selection of the input values. All the results are then collected to generate a statistical characterization of the outcome.

The Monte Carlo method

The Monte Carlo method (MC) [24] is the oldest and most popular sampling approach. It involves random sampling from the space of the random variables ξ according to the given PDFs. The outcome is typically organized as a histogram and the statistics are readily computed from the statistical moments by replacing the integrals in Equation 1.1 with sums over the number of samples. The method has the advantage that it is simple, universally applicable and does not require any modification to the available (deterministic) computational tools. It is important to note that while the method converges to the exact stochastic solution as the number of samples goes to infinity, the convergence of the mean error estimate is slow. Hence thousands or millions of data samples may be required to obtain accurate estimations.

However, the convergence does not directly depend on the number of random variables in the problem. In this form the Monte Carlo methods always give the correct answer, but a prohibitively large number of realizations may be required to accurately estimate responses that have a small probability of occurrence. On the other hand, the convergence of the low order statistics (expected value and variance) require much smaller number of samples.

Latin Hypercube Sampling (LHS)

Several methods have been developed to accelerate the MC approach. One of the most successful is the Latin Hypercube sampling (LHS) approach. LHS is a stratified-random procedure which provides an efficient way of sampling variables from their distributions[25]. The cumulative distribution for each input variable is divided into N equi-probable intervals. A value is selected randomly from each $i - th$ interval and the sampled cumulative probability can be written as[26]:

$$p_i = \frac{1}{N}r_u + (i - 1)N \quad (1.2)$$

where r_u is a uniformly distributed random number ranging from 0 to 1. The N values obtained for each variable are paired randomly with the other variables to construct a sample point in the parameter space. Unlike simple random sampling, this method ensures full coverage of each variable range by optimally stratifying each marginal distribution.

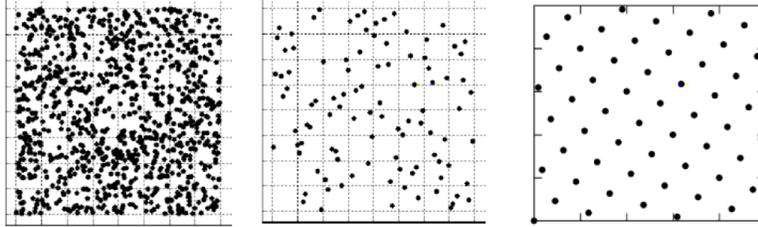


Figure 1.1: Sampling based techniques with two uncertain inputs. (left) Monte Carlo; (center) Latin Hypercube; (right) Lattice based

1.4.2 Quadrature methods

One of the objective of UQ propagation methods is the computation of the statistics of an outcome of interest, such as its expectation and the variance. As shown earlier, these require the evaluation of integrals (over the parameter space) and it is therefore natural to employ conventional numerical integration techniques [27]. Let's consider a problem with one uncertain parameter ξ ; the objective is to compute integrals of $y(\xi)$. A class of quadrature rules are based on interpolating basis functions that are easy to integrate, typically polynomials. The integral is expressed as a weighted sum of the integrand y evaluated in a finite number of locations on the ξ -axis (abscissas). The choice of the polynomial basis defines the weights and the corresponding abscissas. The simplest example is the midpoint rule while quadratures based on equally spaced abscissas include the commonly used trapezoidal and Simpson rules and are, in general, referred to as Newton-Cotes formulas.

Stochastic collocation

Stochastic collocation refers to quadrature methods used to compute integrands of random variables, thus over a stochastic domain. Although Newton-Cotes formulas are applicable in this context, it is usually preferable to consider more general approaches, in which the abscissas are not equally spaced. Gaussian quadratures are popular in the field of uncertainty analysis because of their high accuracy [28]. The most commonly used form of Gaussian quadrature is the Gauss-Legendre integration formula which is based on Legendre polynomials (see Figure 1.2).

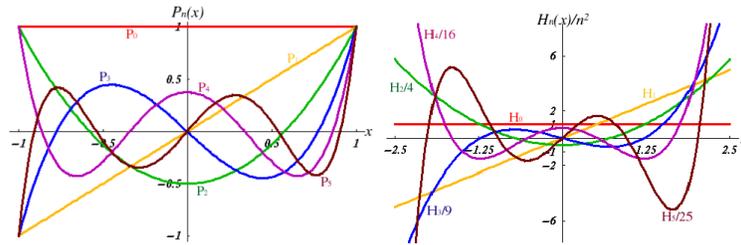


Figure 1.2: Legendre (left) and Hermite (right) polynomials

In practical terms, collocation methods for uncertainty propagation require the evaluation of zeros and weights for a family of orthogonal basis functions; these can be computed and stored in advance. A set of independent computations are performed and the results are combined to obtain the statistics of the output of interest. Collocation can therefore be interpreted as a sampling technique; it retains the main advantage of the Monte Carlo method because it does not require modifications to the existing computational tool. The evaluation of the PDF of the output quantities is somewhat

more complicated for stochastic collocation methods than the computations of the output statistics. The first step is the construction of the polynomial interpolant of the solution in the parameter space. At this point, the interpolant is used as a replacement for the original function [29] and Monte Carlo sampling is used. In the next section we will focus on a particular stochastic collocation techniques based on simplex elements.

1.4.3 The other approaches

In addition to the methods presented here several other methods have been applied especially in the field of structural mechanics. It is also worth mentioning that alternative approaches not based on probabilistic reasoning have been proposed and used with some success. It is not generally clear when probabilistic methods fail or are insufficient; the treatment of epistemic uncertainty remains difficult and possibly the greatest challenge in uncertainty quantification. The choice of the appropriate method to use for a specific application is not obvious. For typical fluid mechanics simulations some common considerations are:

1. expensive function evaluation: sampling based methods are typically not appropriate because they might require several thousand full computations to build the statistics of the outputs;
2. large number of uncertainties: boundary conditions, material properties, geometry specification, etc. introduce many independent input parameters that have to be characterize. Methods that suffer from curse of dimensionality[30] quickly become unfeasible;

3. non-linear system responses: transitions and bifurcations are typical of fluid mechanics, especially for compressible flows. Methods that strictly require a smooth dependency between inputs and outputs can be ineffective.

1.5 Simplex Stochastic Collocation (SSC)

Due to the relatively slow convergence rate of Monte Carlo simulations, other uncertainty quantification methods have been developed based on a polynomial approximation of the response. Stochastic Collocation (SC) is a widely used example of such a method, which is based on sampling Gauss quadrature points and using Lagrangian polynomial interpolation in probability space. However, due to the structured grid of the quadrature points in multiple random dimensions, the spectral convergence of the Stochastic Collocation method reduces significantly with an increasing number of uncertainties.

Here, the Simplex Stochastic Collocation (SSC) method [30, 31] is presented that combines the effectiveness of random sampling in higher dimensions with the accuracy of polynomial interpolation. It also leads to the superlinear convergence behavior of Stochastic Collocation methods and the robustness of Monte Carlo approaches. SSC discretizes the parameter space Ξ using non-overlapping simplex elements Ξ_j from a Delaunay triangulation of sampling points, with $\Xi = \bigcup_{j=1}^{n_e} \Xi_j$, where n_e is the number of elements. In each of the simplexes Ξ_j , the response surface of the quantity of interest, $u(\boldsymbol{\xi})$ as function of the random parameters $\boldsymbol{\xi} \in \Xi$, is approximated by a

polynomial $w_j(\boldsymbol{\xi})$

$$w_j(\boldsymbol{\xi}) = \sum_{m=0}^P c_{j,m} \Psi_{j,m}(\boldsymbol{\xi}), \quad (1.3)$$

with $P + 1$ coefficients $c_{j,m}$ and basis polynomials $\Psi_{j,m}(\boldsymbol{\xi})$. The polynomials are found by interpolation of the samples $v_k = u(\boldsymbol{\xi}_k)$ at the vertexes $\boldsymbol{\xi}_k$ of the simplex elements, with $k = 1, \dots, n_s$, where n_s is the number of samples. For higher degree interpolation a stencil of sampling points $v_{k_j,l}$ at the vertexes $\boldsymbol{\xi}_{k_j,l}$ of surrounding simplexes is constructed, with $l = 0, \dots, N$ and $k_{j,l} \in \{1, \dots, n_s\}$. The polynomial coefficients $c_{j,m}$ are then given by

$$\begin{bmatrix} \Psi_{j,0}(\boldsymbol{\xi}_{k_{j,0}}) & \Psi_{j,1}(\boldsymbol{\xi}_{k_{j,0}}) & \cdots & \Psi_{j,P}(\boldsymbol{\xi}_{k_{j,0}}) \\ \Psi_{j,0}(\boldsymbol{\xi}_{k_{j,1}}) & \Psi_{j,1}(\boldsymbol{\xi}_{k_{j,1}}) & \cdots & \Psi_{j,P}(\boldsymbol{\xi}_{k_{j,1}}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_{j,0}(\boldsymbol{\xi}_{k_{j,N}}) & \Psi_{j,1}(\boldsymbol{\xi}_{k_{j,N}}) & \cdots & \Psi_{j,P}(\boldsymbol{\xi}_{k_{j,N}}) \end{bmatrix} \begin{pmatrix} c_{j,0} \\ c_{j,1} \\ \vdots \\ c_{j,P} \end{pmatrix} = \begin{pmatrix} v_{k_{j,0}} \\ v_{k_{j,1}} \\ \vdots \\ v_{k_{j,N}} \end{pmatrix}, \quad (1.4)$$

with $N \geq P$. The robustness of the approximation is guaranteed by using a limiter approach for the local polynomial degree p_j , based on the extension of the Local Extremum Diminishing (LED) concept to probability space. This ensures that no overshoots are present in the response interpolation in each of the elements Ξ_j

$$\min_{\Xi_j} (w_j(\boldsymbol{\xi})) \geq \min_{\Xi_j} (u(\boldsymbol{\xi})) \wedge \max_{\Xi_j} (w_j(\boldsymbol{\xi})) \leq \max_{\Xi_j} (u(\boldsymbol{\xi})), \quad (1.5)$$

for $j = 1, \dots, n_e$. The initial samples consist are located at the parameter range extrema and one at the nominal conditions, see Figure 1.3a for a two-dimensional example. The discretization is adaptively refined by calculating a refinement measure based on a local error estimate in each of the simplex elements. A new sampling point is then added randomly in the simplex



(a) Refinement of the initial mesh

(b) Refined mesh for 17 samples

Figure 1.3: Simplex Stochastic Collocation discretization of a two-dimensional probability space.

with the highest measure and the Delaunay triangulation is updated. The sample is confined to a sub-domain of the simplex to ensure a good spread of the sampling points, see Figure 1.3a. The refinement to $n_s = 17$ samples, shown in Figure 1.3, leads to a super-linear convergence by increasing the polynomial degree p_j with the increasing number of available samples n_s . The sampling procedure is stopped when a global error estimate reaches an accuracy threshold.

In the wind turbine simulations and other large-scale problems, it is possible that one of the deterministic computations for a specific sample of the random parameters does not converge or gives an unrealistic result. For the Stochastic Collocation method such a failure of one of the quadrature samples would be a serious problem in computing statistical moments. On the other hand, this situation forms no obstacle for SSC owing to the flexibility of the randomized sampling. It is handled by introducing a check of the correct execution of the samples into the algorithm. If an unconverged sample is detected, then the failed sample computation is automatically restarted for another randomly sampled point in the refined simplex element. In the

analysis performed in this paper, this has proven to be an effective approach for dealing with erroneous samples, which shows the flexibility of the SSC method in complex computational problems.

1.6 Case study: Analysis under Uncertainty of a Wind Turbine

Wind turbine reliability plays a critical role in the long-term evolution of wind-based energy generation. The computational assessment of failure probability or life expectancy of turbine components is fundamentally hindered by the presence of large uncertainties in both the environmental conditions and blade geometry and structure. Rigorous quantification of the impact of such uncertainties can fundamentally improve the state-of-the-art in computational predictions and, as a result, provide aid in the design of more cost-effective devices.

The present study is the first step of a comprehensive analysis of wind turbine performance under uncertainty. It will be constructed a multi-physics low-order model EOLO that includes aerodynamic predictions, comprehensive structural analysis and acoustic estimation. There will be identified three sources of uncertainty, namely wind condition, insect contamination and manufacturing tolerances, and it will be estimated their effect on aerodynamic performance and noise. Specifically, we demonstrate how the present uncertainties lead to a general decrease in performance with respect to the nominal (design) scenario. This penalization is also compounded with a likely variation in noise. These results indicate that design and optimization steps

should include a comprehensive estimation of the uncertainties in order to achieve robust performance, this framework will be presented in chapter 4. An additional objective of this section is to compare Latin Hypercube Sampling and Stochastic Simplex Collocation for propagating uncertainties in complex computational models. Both methods outperform classical Monte Carlo but it will be shown that the SSC approach leads to stable statistics requiring only a few dozen EOLO simulations.

1.6.1 EOLO: a multi-physics low-order model for wind turbines

Wind turbines are multi-physics devices in which the aerodynamic performance, the structural integrity of the blades, the energy conversion toolbox and the acoustic impact have to be carefully examined to achieve an effective design. Each one of these aspects introduces considerable hurdles for detailed simulations. The aerodynamic performance is dominated by the design of the blade cross-sections. The sections are typically laminar-flow airfoils use to reduce the overall drag. The flow characterization is complicated by the need to predict laminar/turbulent transition under a variety of clean and perturbed wind conditions, the inherent angle of attack variability associated to rotation, the presence of dynamic stall, aeroelasticity, etc. In spite of the development of advanced computational fluid dynamic tools that can predict with reasonable accuracy the aerodynamic performance of rotors [32], the computations remain extremely expensive and often rely on simple models to capture important effects, such as transition, and are gen-

erally not considered to be predictive for extreme events such as stall. In this work, we focus on building a flexible computational infrastructure based on low-fidelity models that are connected together in a *matlab* environment called **EOL0**. There are two main advantages resulting from this choice: *i*) control and flexibility in using different models developed for capturing complex phenomena, *ii*) low computational cost. It is the second aspect that fundamentally enables us to perform analysis under uncertainty.

In the following we introduce the various computational tools that are used to perform the deterministic analysis. The uncertainty quantification methodologies are described in the next section.

Aerodynamic analysis

The geometrical description of the turbine blades are based on the specification of three airfoils at the root, mid-span and tip. Simple linear interpolation is used to construct the geometry at the other cross-sections and the local aerodynamic performance (two-dimensional analysis) is carried out using a potential flow method with interactive viscous correction. The tool we used is **Xfoil** [33] which includes a model for boundary layer transition based on the e^N method. **Xfoil** is used to determine the aerodynamic force coefficients polars in a range of angle of attacks from -15° to 25° to cover the range of incident angles experienced during a full rotation. **Xfoil** is not expected to be accurate in the prediction of stall, because of the presence of extensive flow separation and possibly unsteady effects.

Hence a correction to the polar curve is introduced, based on Viterna[34]

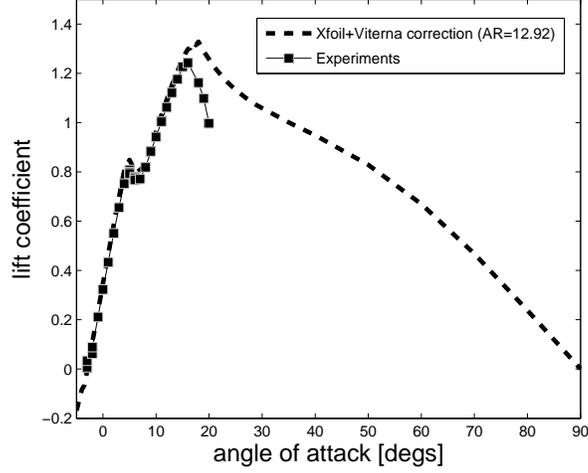


Figure 1.4: NREL s827: $M=0.1$, $Re=3e6$, free transition

and Corrigan models which provide a correction of the lift and drag coefficient at high angle of attack. A final correction to the aerodynamic coefficients is employed due to the presence of finite-span effects. Here we use a modification based on the Lanchester- Prandtl theory

$$C_L = C'_L; \quad C_D = C'_D + \frac{Cl^2}{\pi AR}; \quad \alpha = \alpha' + \frac{Cl}{\pi AR} \quad (1.6)$$

where C_L and C'_L , C_D and C'_D , α and α' are the finite and infinite span airfoil lift coefficients, drag coefficients and angles of attack respectively and AR is the aspect ratio of the wind blade.

The Viterna model estimates the lift and drag coefficients (when $\alpha > \alpha_s$) as follows

$$C_L = \frac{C_{Dmax}}{2} \sin 2\alpha + K_L \frac{\cos \alpha^2}{\sin \alpha}; \quad C_D = C_{Dmax} \sin \alpha^2 + K_D \cos \alpha \quad (1.7)$$

$$K_L = (C_{LS} - C_{Dmax} \sin \alpha_s \cos \alpha_s) \frac{\sin \alpha_s}{\cos \alpha_s^2}; \quad K_D = \frac{C_{DS} - C_{Dmax} \sin \alpha_s^2}{\cos \alpha_s} \quad (1.8)$$

where α_s is the finite span airfoil stall angle, C_{LS} and C_{DS} are the finite span airfoil lift and drag coefficient at stall angle and C_{Dmax} is the finite span airfoil maximum drag coefficient

$$\begin{aligned} C_{Dmax} &= 1.11 + 0.018AR & AR \leq 50 \\ C_{Dmax} &= 2.01 & AR > 50 \end{aligned} \tag{1.9}$$

The models described above are included in a tool called **Viterna** which collects the polars from **Xfoil** and introduces stall and finite-span corrections. The present predictions of the lift curve are reported for the NREL S827 airfoil in Fig.1.4.

Structural analysis

Fluid structure interactions play an important role in the determination of the structural integrity of the turbine blades and in the overall aerodynamic performance.

The geometrical description of the blade is used as a starting point to define span-varying properties relevant to its composite structure. The NREL **PreComp** [35] computes cross-coupled stiffness, inertia and offsets of the blade shear center, tension center, and center of mass with respect to the blade pitch axis. These quantities are then used to determine a low-order model for the rotor, tower and drivetrain shaft. Specifically, the characteristics of a rotating-beam equivalent to the blade are computed using NREL **BModes** [36], a finite-element code that evaluates the deformation modes.

Performance analysis

The `Viterna` corrected polars at certain nodes along the span, the flapwise and edgewise `Bmodes` modal shapes and the `PreComp` properties are then used as input to NREL `FAST` [37] (Fatigue, Aerodynamics, Structures, and Turbulence) which is a comprehensive aeroelastic simulator capable of predicting both the extreme and fatigue loads of two- and three-bladed horizontal-axis wind turbines.

This code is based on the NREL `AeroDyn` [38] solver, an element-level wind-turbine aerodynamic analysis routine. It requires information on the status of a wind turbine from the dynamics analysis routine and a *wind file* describing the atmospheric conditions. It returns the aerodynamic loads for each blade element to the dynamics routines.

Wind conditions

The aerodynamic performance of wind turbines is dominated by the wind conditions. Atmospheric boundary layers are subject to large variability in wind direction and intensity with largely unsteady dynamics and frequent gusts. In `EOL0` we generate realistic wind conditions using the NREL `TurbSim`[39] tool, which constructs a stochastic inflow with a precisely specific velocity fluctuation spectrum.

Acoustic analysis

The NREL prediction of aeroacoustic noise is based on six different noise sources (Fig. 1.5) that are assumed to independently generate their own noise

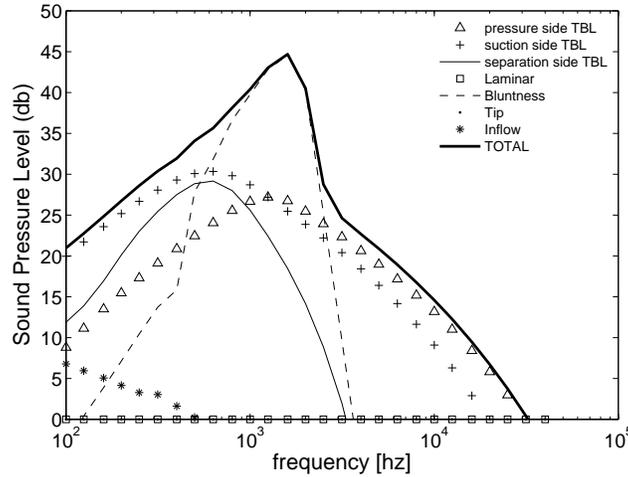


Figure 1.5: Breakdown of the noise generated by a 50kw wind turbine at a microphone located at $(x,y,z)=(-20m,0m,0m)$

signature. The assumption of independence is based on the idea that the mechanisms for each noise source (namely turbulent boundary layer trailing edge, separating flow, laminar boundary layer vortex shedding, trailing edge bluntness vortex shedding, and tip vortex formation [40, 41]) are fundamentally different from each other or occur in different locations along a turbine blade, such that they do not interfere with one another.

the EOLO matlab script

The various tools briefly described in the previous subsections are glued together in a multi-physics simulation process using `matlab`. The overall driver script, `EOLO` handles the transfer of information between the various tools and then collects the final outputs and computes statistics.

A flowchart of the process is reported in Fig. 1.6; it is clear that mod-

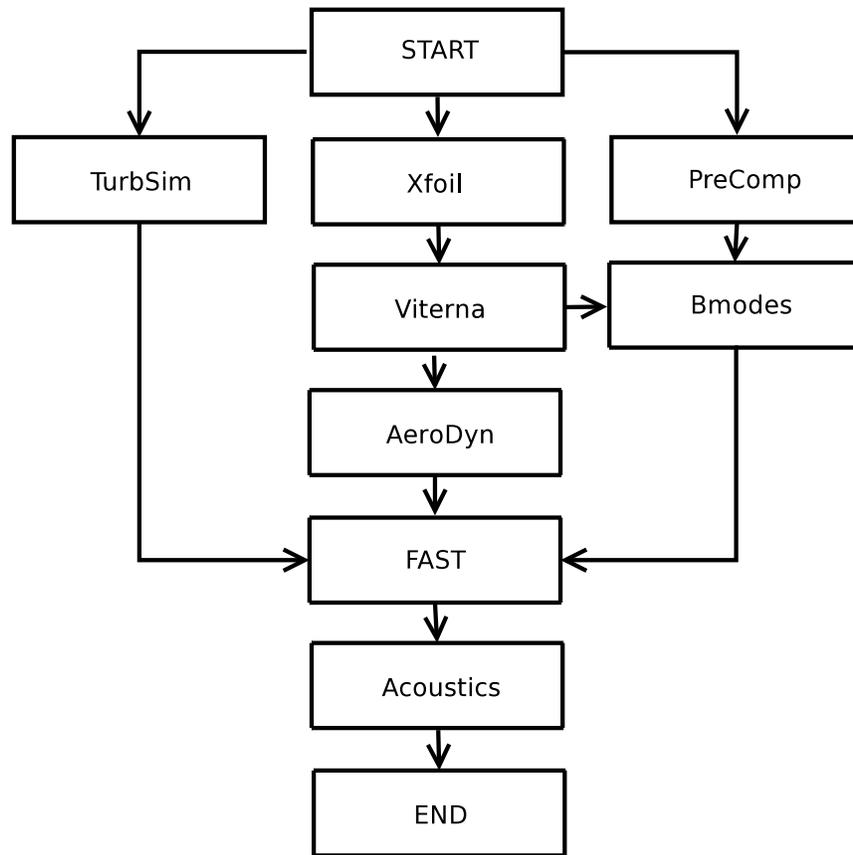


Figure 1.6: EOLO flowchart

ifications to the framework can be handled in a simple way, for example substituting the aerodynamic performance evaluation module (Xfoil and Viterna) with a computational fluid dynamic solver. EOLO also provides a unique interface for the entire process (from inputs to outputs) that is directly connected to the uncertainty quantification tools.

1.6.2 The AOC 15/50

The AOC 15/50 is a downwind turbine, i.e. its blades rotate downwind of the drive train assembly. Furthermore, it has no active yaw control and depends on its blades to track the wind. This wind turbine is the evolution of the rugged and reliable Enertech E44, many of which were installed in the 1980's and are still running today. Independent analysis and testing at NREL, the Netherlands Energy Research Foundation (ECN), RISO Laboratory in Denmark, the Atlantic Wind Test Site (AWTS) on Prince Edward Island and other sites around the world verify that the AOC 15/50 wind turbine generators are very reliable in even the harshest weather conditions. The AOC 15/50 is designed for simplicity to minimize maintenance requirements and to be able to safely operate in normal and extreme conditions. The principal characteristics of the AOC 15/50 can be found in Table 1.1.

1.6.3 Analysis under Uncertainty

EOL0 is essentially *deterministic*: once the wind-turbine configuration and other input conditions are specified, the solution is uniquely determined without vagueness. On the other hand, when uncertainties are present, the results have to be expressed in a non-deterministic fashion either probabilistically or as ranges of possible outcomes. In this work we focus on the former, and describe the uncertainties as random variables. At this point the computations become *probabilistic* in nature and it is necessary to propagate the input variability into the output of interest (quantity of interest, QoI). The

CHAPTER 1. UNCERTAINTY QUANTIFICATION

Table 1.1: AOC 15/50 wind turbine, principal characteristics

Type	Grid Connected
Configuration	Horizontal Axis
Axis Rotor Diameter	15 m
Centerline Hub Height	25 m
Rated Electrical Power	50 kW @ 11.3 m/s
Cut in [-]	4.6 m/s
Shut down (high wind)	22.4 m/s
Type of Hub	Fixed Pitch Rotor
Diameter	15 m
Swept Area	177 m^2
Number of Blades	3
Rotor Solidity	0.077
Rotor Speed @ rated wind speed	65 rpm
Location Relative to Tower	Downwind
Cone Angle	6°
Tilt Angle	0°
Rotor Tip Speed	51 m/s @ 60 Hz
Design Tip Speed	6.1
Length	7.2 m
Material	Epoxy /glass fibre
Airfoil (type)	NREL, Thick Series, modified
Twist	7°
Blade Weight	150 kg approximate
Yaw	Free, rotates 360 degrees

approach we follow here is strictly non-intrusive and the existing tools are used without modifications, but the solution - or more precisely, their probability distributions - are constructed performing an ensemble of deterministic analysis. It is important to note that the first step in any uncertainty quantification procedure is the identification of the sources of uncertainties. We focus on wind conditions, blade manufacturing tolerances and insect contam-

ination. Indeed we gather information from literature regarding each one of these sources and perform analysis that assesses their impact on both aerodynamic performance and noise. In order to assess the performance of the uncertainty propagation process we compare two methodologies that aim at characterizing statistically the QoIs using a low number of deterministic solutions compared to classical Monte Carlo schemes. The two approaches are the Latin Hypercube Sampling (LHS) and the Stochastic Simplex Collocation (SSC), described in the previous sections.

Data assimilation: uncertain meteorological conditions

The energy produced by a wind turbine is usually expressed as an annual average. Since production falls off dramatically as the wind speed drops, most of the time the wind turbine is producing well below its expected rate [42]. It is important to characterize the wind turbine behavior resulting from the measured wind variability to assess the effective performance.

For land based turbines, the wind speed distribution is usually approximated by a Weibull fit [43]. As an example, Downey [44] extracted data from the database <http://winddata.com> of eight sites that have wind speed measurements above 60 m in height.

Following the same approach we extracted nominal wind speed, turbulence intensity and direction data at a site (Acqua Spruzza, Italy) where a wind turbine farm was built by ENEL S.p.A. to evaluate the performance of commercial medium-sized turbines operating in complex terrain and very hostile climate. A large collection of wind measurements is summarized in

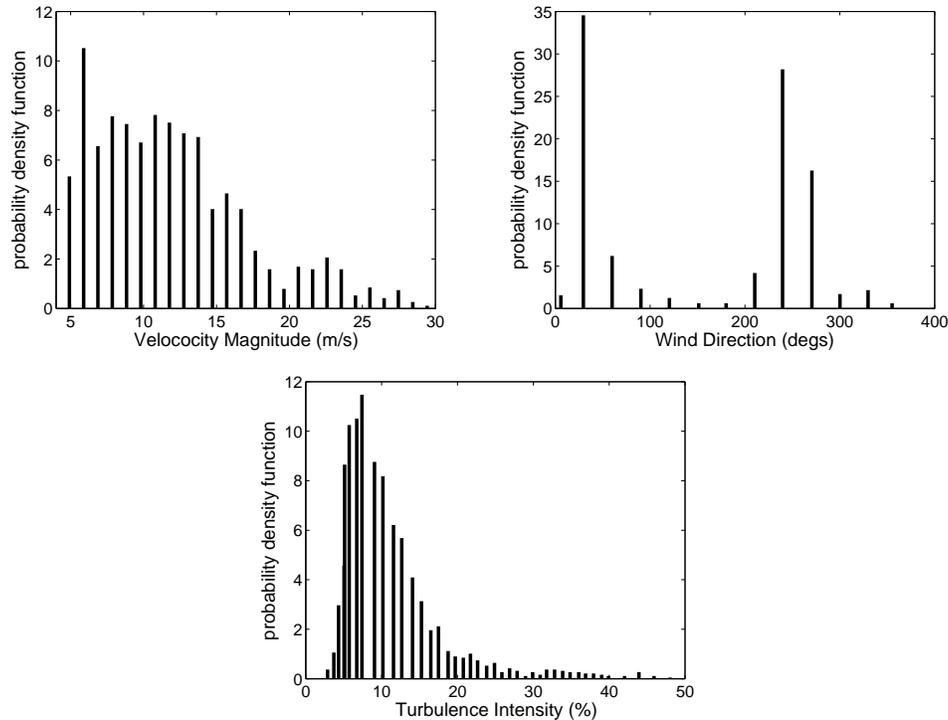


Figure 1.7: Wind speed, direction and turbulence intensity at the Acqua Spruzza, Italy site. The data is reported in terms of empirical probability distributions scaled from 40 to 24 meters.

Fig. 1.7 in terms of wind speed and direction and turbulence intensity. The histograms of these three random variables are used directly as input for the uncertainty propagation methods described previously, after being converted into continuous probability density functions (for each of the input variables) via linear interpolation. Note that no information regarding the correlation of the three random variables is available, and therefore we assume that the inputs are independent.

The wind data readily available provide an estimate of the wind speed at a certain height. To construct the wind conditions at the actual rotor hub

height (≈ 24 meters) we use a classical [45] scaling law:

$$\frac{U(z_2)}{U(z_1)} = \frac{\ln(z_2/z_0) - \Psi(Z_2/L)}{\ln(z_1/z_0) - \Psi(Z_1/L)} \quad (1.10)$$

where L is the Monin-Obukhov length [m], U is the wind speed [m/s], z is the height [m], z_0 is the terrain roughness [m] and Ψ is the stability function [42]. Eq. (9) uses a logarithmic velocity profile dependent on terrain roughness z_0 plus an atmospheric stability correction Ψ dependent on the ratio of height, z , and Monin Obukhov length, L . Turbulence may be produced by shear (speed differences) or by buoyancy (density differences); the length L is the height at which shear and buoyancy produce the same amount of turbulent kinetic energy. The Monin-Obukhov length L must be estimated. When the wind speed is sufficiently high (above $6m/s$) thermal effects play no role above land, and neutrality may be assumed [46], $\Psi = 0$. In this case the wind speed follows a logarithmic profile determined by terrain roughness z_0 and Eq. (9) reduces to

$$\frac{U(z_2)}{U(z_1)} = \frac{\ln(z_2/z_0)}{\ln(z_1/z_0)} \quad (1.11)$$

The roughness of the terrain z_0 may be estimated with the Petersen classification [47] or from measurements of turbulence intensity I at some reference height z_r , using:

$$z_0 = z_r \exp\left(\frac{-1}{I(z_r)}\right) \quad (1.12)$$

Data assimilation: insect contamination

Several studies on wind turbines [48, 49, 50, 51] and fixed wings [52, 53] illustrate the effect of insect and dirt contamination on the overall aerodynamic performance. Insects are present in the lower layer of the atmosphere,

with a density rapidly decreasing from ground level to 500 ft. Hardy and Milneceite [54] found that the morphology of insects is a function of the altitude and that estimation of the actual contamination depends on the operating conditions. In wind-turbines the effect of contamination can be particularly strong when the blade cross-sections are designed to support mostly laminar flows. The presence of insect contamination produces boundary layer disturbances that can lead to early transition to turbulence with a deterioration of the aerodynamic performance. This is the motivation for including insect contamination as a leading cause of uncertainty in the analysis of wind turbines.

Crouch *et al* [55] studied experimentally the effects of surface protrusions (steps) on the transition to turbulence in boundary layers. They also modified the e^N method to capture the observed transition modifications, via a reduction of the critical N-factor:

$$N_{crit} = N_{crit^0} - \Delta N_{crit} \left(\frac{h}{\delta^*} \right) \quad (1.13)$$

where h is the height of the step (i.e. the accumulated insect height)[m], δ^* is the boundary layer displacement thickness at the step location [m], ΔN_{crit} accounts for the local change in the stability characteristics at the step[-] and N_{crit^0} is the clean value of the critical n-factor[-].

In this work we assume that the insect impact produces a roughness that leads to a possible modification of the N-factor. We consider three independent variables describing the N-factor ranging from clean conditions ($N_{crit} = 9$) to transition bypass ($N_{crit} = 1$) at the root, midspan and tip

sections.

Data assimilation: manufacturing errors

There is a general agreement that airfoil shape, twist and chord length imperfections are detrimental to aerodynamic performance, but only limited quantitative data is available in the open literature about their origin and quantitative effects. Loeven and Bijl [56] used a Polynomial Chaos Framework for the quantification of airfoil geometrical uncertainties. Ilinca, Hay, and Pelletier [57] treat shape sensitivities of unsteady laminar flow around a cylinder in ground proximity. Etienne *et al* [58] investigated shape sensitivities of flexible plates in a flow domain. Gumber, Newman, and Hou [59] included first order moments in robust design optimization of a 3D flexible wing with uncertain wing geometry. The geometry of a manufactured wind turbine airfoil is generally different from the nominal design mainly because of manufacturing tolerances. It is generally difficult to characterize probabilistically the effect of these tolerances; in this work we focus on errors associated with the protusion process, where the blade is constructed as a sequence of cross-section. We assume that the twist of the blade (the section orientation with respect to a nominal plane) is imprecise. As before, we assume that we can describe the uncertainty using three independent parameters (with uniform probability distributions ranging from -2° to 2°) associated to the twist at the root, the midspan and the tip section.

Certification: uncertain meteorological conditions

The AOC 15/50 has been investigated using the uncertain meteorological conditions of Figure 1.7. In this case EOL0 is driven by the SSC routines and the uncertainties are injected through Turbsim; the statistics are constructed performing an ensemble of deterministic analysis [60]. For reasons of economy, the wind history during the turbine’s approximately twenty year life is reduced to 10 minute periods (or load cases) at each wind speed [42]. The latitude chosen for the turbulence model is 41 degrees, matching the data extracted from the Acqua Spruzza site. The Von Karman spectral model for the meteorological boundary conditions has been chosen in this application, assuming neutral atmospheric condition [61, 62]. The AOC 15/50 deterministic conditions chosen as reference in this work are illustrated in Table 1.2.

Table 1.2: AOC 15/50 wind turbine, deterministic conditions

Wind speed	6 m/s
Turbulence intensity	5%
Wind direction	0°
Latitude	41°
Working time	10 min
Rotor speed	55 rpm

The Monte Carlo samples on the response surface obtained by the simplex reconstruction are shown in Figure 1.8; the reader can notice that the samples follow the input distribution of Figure 1.7.

A three color (red to blue) map has been introduced to relate the samples to the effective value of the power coefficient in the domain: the red points

correspond to high power extracted by the wind turbine. The map reveals that improved efficiency is achieved for moderate wind speeds (5-12 m/s) and low turbulence levels (2-10 percent), while other conditions lead to decreased performance.

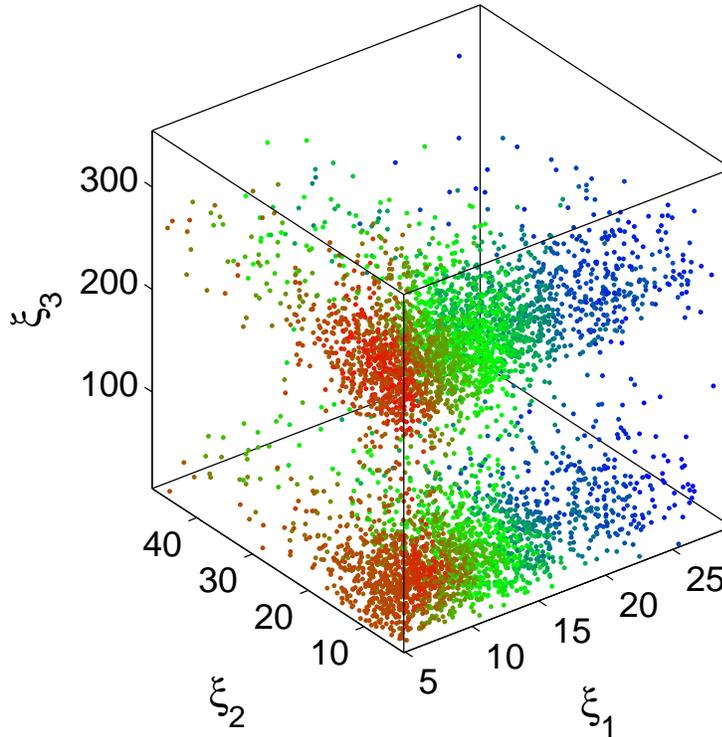


Figure 1.8: Monte Carlo samples for meteorological conditions.

Since the cut-off speed of the AOC 15/50 is 22.4 m/s, placing the wind turbine in the site of interest would likely lead to off-design operating conditions: the average power coefficient drops from 0.4596 to 0.2776.

The resulting cumulative probability distribution functions (CDF) for the power coefficient and the sound pressure level as a result of the varying

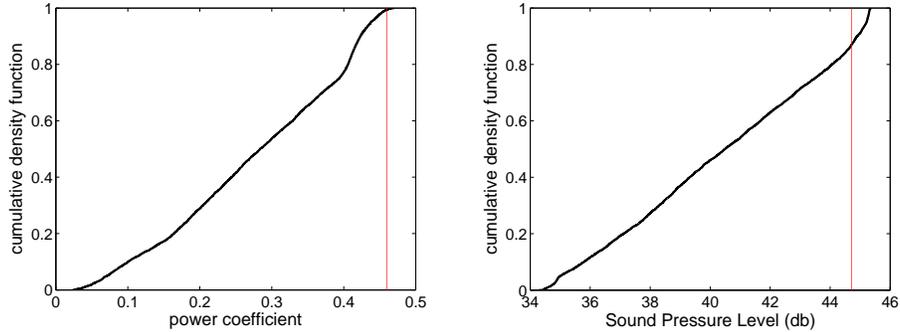


Figure 1.9: Meteorological conditions: cumulative density function of the power coefficient and Sound Pressure Level. The red lines represent the deterministic conditions.

meteorological conditions are shown in Figure 1.9. The output CDFs show approximately uniform distributions, in contrast with the input density for the meteorological conditions of Figure 1.7. The probability distributions fall completely below the deterministic characteristics of the wind turbine (Table 1.2) given by the vertical lines. The uncertain output for the power coefficient ranges from approximately 0 to the deterministic value of approximately 0.45. The sound pressure level varies uniformly between 34 to 45 dB. These results show that the realistic uncertainty in the wind speed, direction, and turbulence intensity has a large impact on the wind turbine performance.

The convergence of the mean and standard deviation of the power coefficient and sound pressure level is shown in Figures 1.10 and 1.11 up to 70 samples in the SSC discretization. The mean values of the two outputs show fast convergence in the first 20 samples to a value significantly lower than the deterministic value. Increasing the number of deterministic solves to 70 does not significantly change the mean value. This is confirmed by the decreasing

CHAPTER 1. UNCERTAINTY QUANTIFICATION

error estimate intervals with an increasing number of samples. The higher moment of the standard deviation shows, as expected, a slower convergence up to 40 samples with a relatively larger error estimate margin.

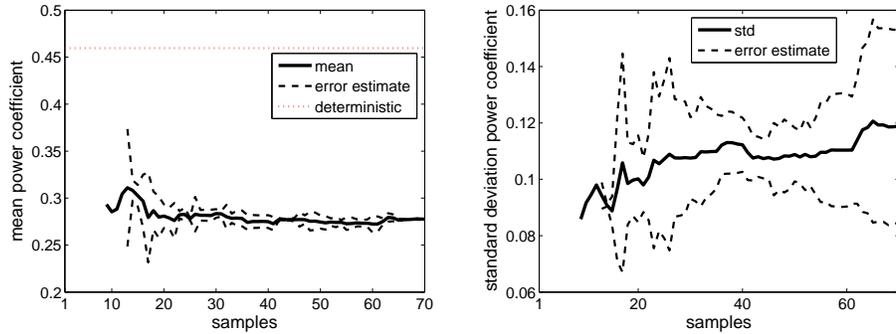


Figure 1.10: Meteorological conditions: convergence histories of the mean, variance and error of the power coefficient

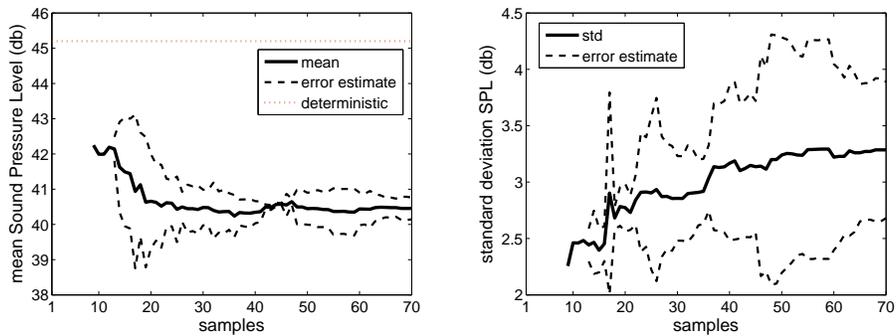


Figure 1.11: Meteorological conditions: convergence histories of the mean, variance and error of the Sound Pressure Level

The values for the mean and standard deviation at 70 samples are also compared to the nominal clean configuration in Table 1.3. Due to the uncertainty the mean power output almost halves, while the sound level is only moderately lower than the deterministic values. This demonstrates that the

uncertain meteorological conditions result in a significant reduction of the power performance of the wind turbine, which shows the need to optimize the wind turbine design under uncertain operating conditions. The relatively large coefficients of variation in the power coefficient and the sound pressure level up to 42.8% and 8.1%, respectively, are also reported.

Table 1.3: Analysis under uncertain meteorological conditions

Objective	Mean	Standard deviation	Coefficient of variation	Clean configuration
# Power Coefficient [-]	0.2776	0.1189	0.4283	0.4596
# Sound Pressure Level [db]	40.4530	3.2853	0.0812	44.711

Certification: insect contamination

The AOC 15/50 is investigated using the uniform distributions of section 1.6.3. In this case `EOL0` is driven by the SSC routines and the uncertainties are injected through the aerodynamic coefficients computed in `Xfoil`.

The Monte Carlo samples on the response surface obtained by the simplex reconstruction are shown in Figure 1.12; the output samples have a rather uniform distribution in the probability space.

The colormap reveals that higher values of the n-critical factor (e.g. lower contamination) at tip region, ξ_3 primarily, as well as at mid-span region, ξ_2 , lead to better performance: this can be justified due to the highest contribution of the outer airfoils to the mechanical torque at the shaft.

This analysis illustrates a reduction of up to 16% in the power coefficient (Figure 1.13) due to the insect contamination, while in the literature an effect of up to 50% has been reported [48, 49]. This difference might be due to the present approach used to characterize the effect of the insect contamination.

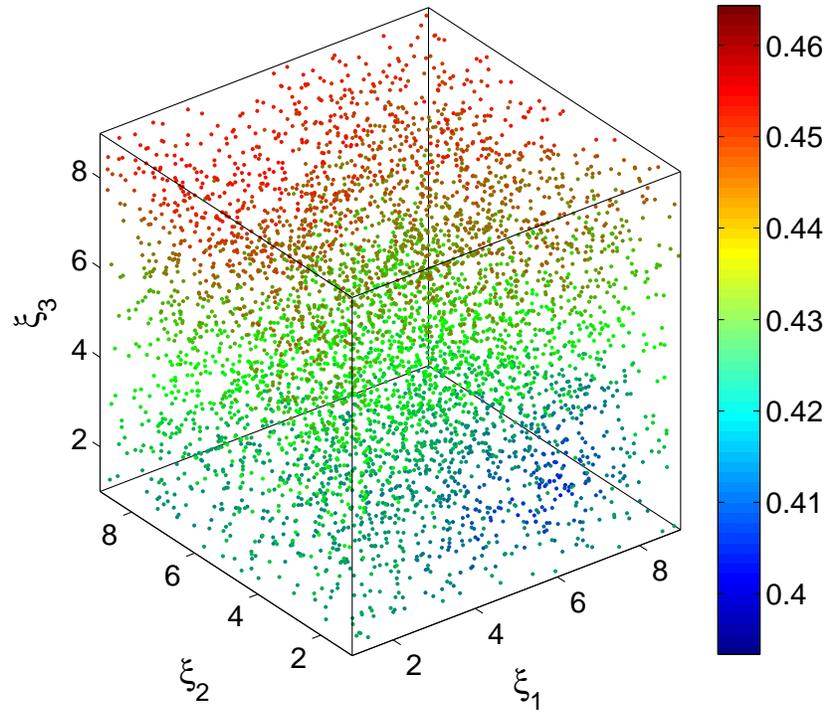


Figure 1.12: Monte Carlo samples for insect contamination

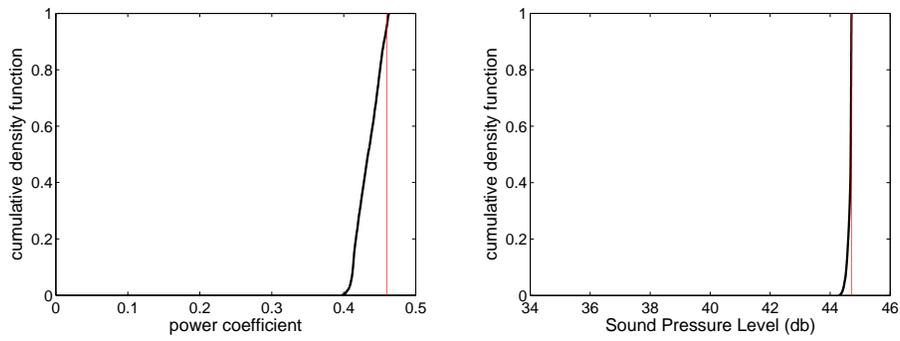


Figure 1.13: Insect contamination: cumulative density function of the power coefficient and Sound Pressure Level

CHAPTER 1. UNCERTAINTY QUANTIFICATION

The variation of the perceived level of noise due to this source of uncertainty can be neglected.

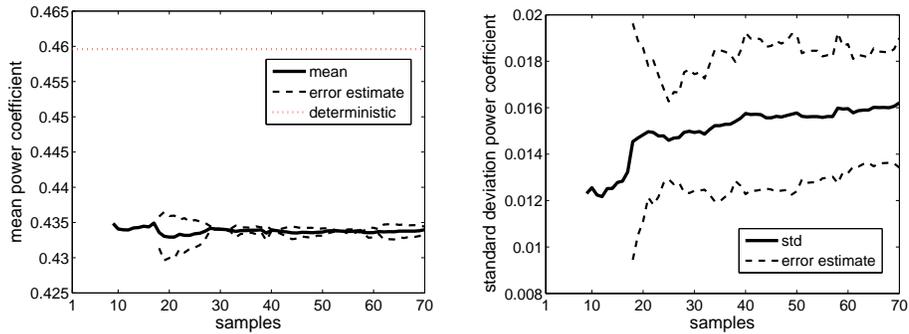


Figure 1.14: Insect contamination: convergence histories of the mean, variance and error of the power coefficient

Figure 1.15 and Figure 1.16 show the SSC convergence of the mean and the standard deviation of the output of interest. The error estimate is lower under uncertain meteorological conditions, therefore a smaller number of simplex points could have been used.

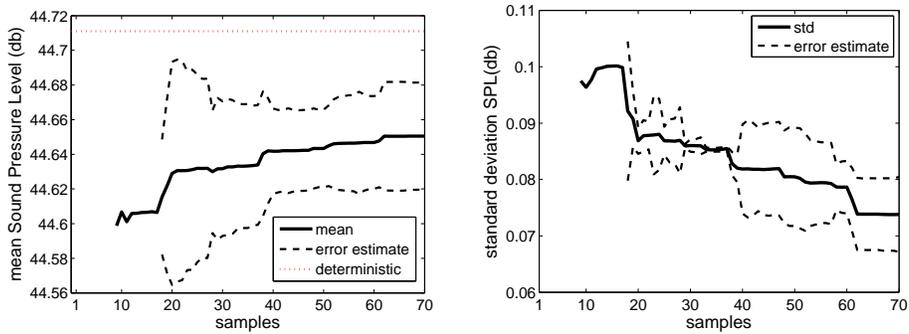


Figure 1.15: Insect contamination: convergence histories of the mean, variance and error of the Sound Pressure Level

The mean, standard deviation and coefficient of variation of the analysis

under insect contamination are summarized in Table 1.4.

Table 1.4: Analysis under insect contamination

Objective	Mean	Standard deviation	Coefficient of variation	Clean configuration
# Power Coefficient[-]	0.4340	0.0162	0.0373	0.4596
# Sound Pressure Level[db]	44.6505	0.0738	0.0017	44.711

Certification: manufacturing errors

The AOC 15/50 is investigated using the uniform distributions of section 1.6.3.C. In this case EOLO is driven by the SSC routines and the uncertainties are injected through the geometry pre-processor. The Monte Carlo samples on the response surface obtained by the simplex reconstruction are shown in Figure 1.12; similar to the insect contamination case, the samples have a uniform distribution in the probability space. The colormap reveals that decreasing the twist in the mid-span region, ξ_2 , leads to better performance: this can justify a novel robust shape optimization involving the twist distribution.

In this framework we were able to reduce the power coefficient by up to 7% (Figure 1.17) with negligible change in the perceived noise.

Figure 1.18 and Figure 1.19 show the SSC convergence of the mean and standard deviation of the output of interest, revealing that 30 simplex points would have been enough for this analysis.

The mean, standard deviation and coefficient of variation of the analysis under manufacturing errors are summarized in Table 1.5.

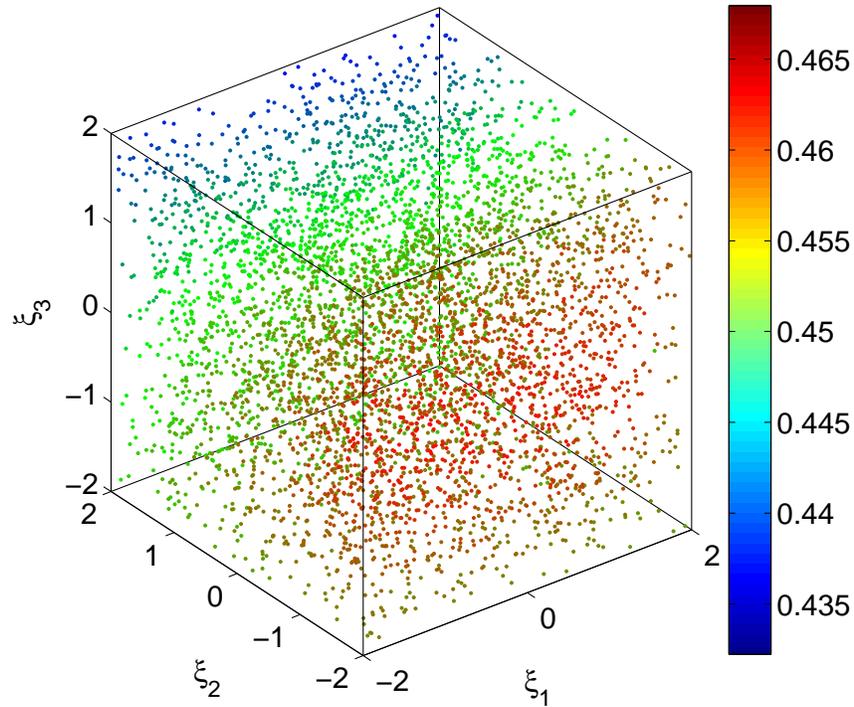


Figure 1.16: Monte Carlo samples for manufacturing errors

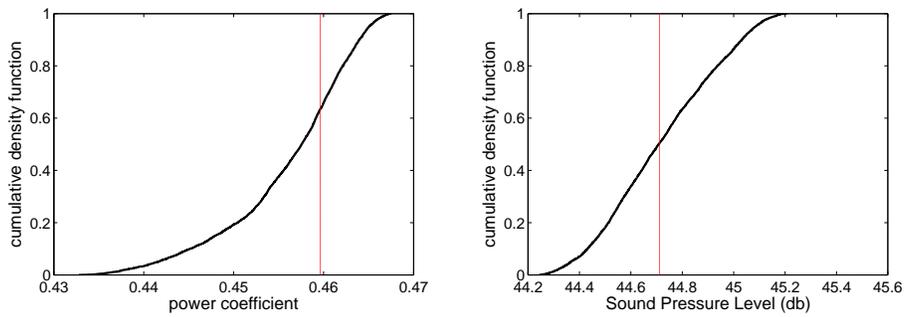


Figure 1.17: Manufacturing errors: cumulative density function of the power coefficient and Sound Pressure Level

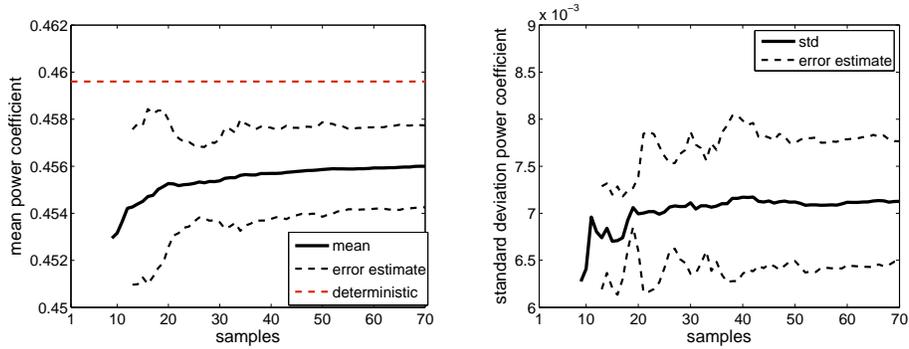


Figure 1.18: Manufacturing errors: convergence histories of the mean, variance and error of the power coefficient

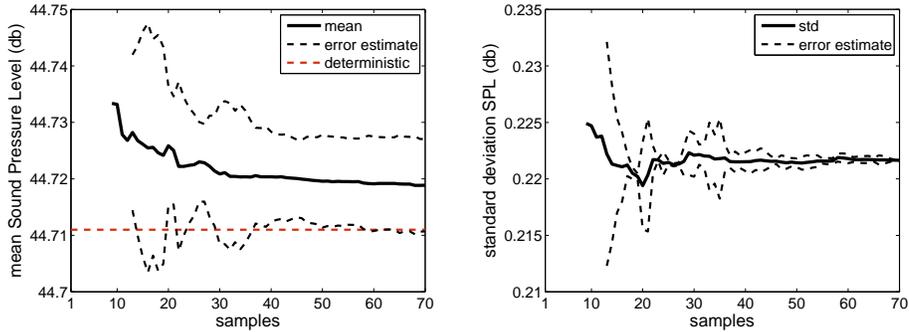


Figure 1.19: Manufacturing errors: convergence histories of the mean, variance and error of the Sound Pressure Level

Table 1.5: Analysis under manufacturing errors

Objective	Mean	Standard deviation	Coefficient of variation	Clean configuration
# Power Coefficient[-]	0.4560	0.0071	0.0156	0.4596
# Sound Pressure Level[db]	44.7189	0.2216	0.0050	44.711

Uncertainty propagation: SSC vs LHS

In this last part of the analysis under uncertainty we want to quantify the effect of the uncertainty propagation process by comparing SSC and LHS. The convergence histories for the mean and standard deviation of the power

coefficient and sound level are more closely examined in Figures 1.20 and 1.21.

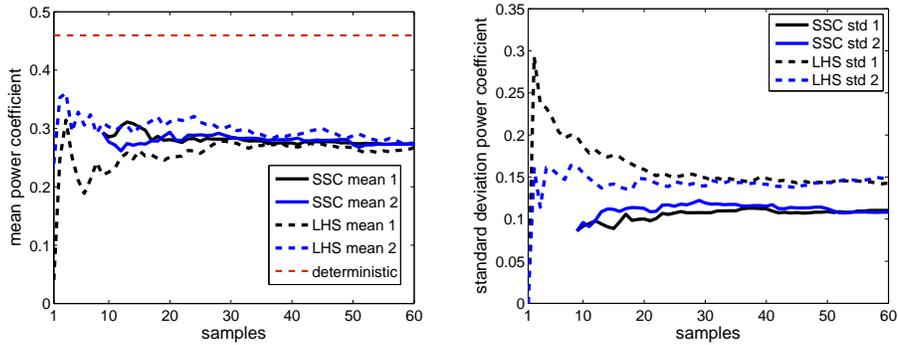


Figure 1.20: Meteorological conditions: comparison of the mean and variance of the power coefficient for two different SSC and two different LHS

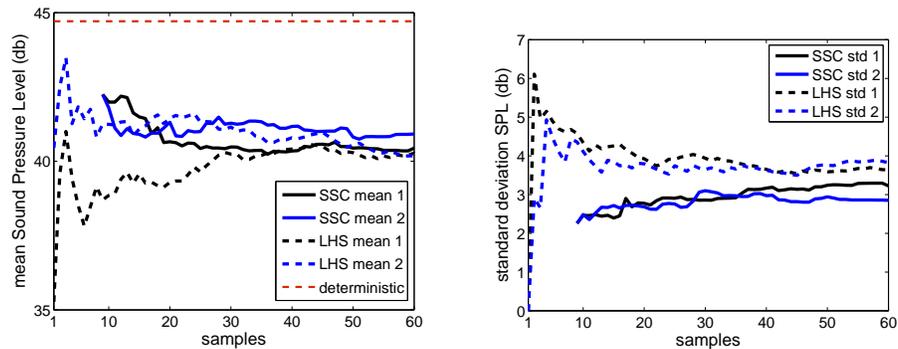


Figure 1.21: Meteorological conditions: comparison of the mean and variance of the Sound Pressure Level for two different SSC and two different LHS

The results of SSC and LHS are shown for two independent ensembles to illustrate the relative sensitivity as both methods rely on random sampling. This is clearly visible in the LHS results for the power in Figure 1.20 given a small number of samples (up to 40). For a larger sample size the two LHS results show improved agreement. The variations in the SSC results for the power coefficient are smaller than for LHS.

For the mean power output, the two methods show good agreement. The prediction of the standard deviation by LHS and SSC show a slightly different value, because we hypothesize that more than 60 samples are required to obtain convergence. The results for the sound level in Figure 1.21 show a relatively larger variation between the SSC runs and also gives a larger standard deviation output for LHS than for SSC.

1.7 Lessons learned

In this chapter it was shown that stochastic collocation methods, in particular the Simplex Stochastic Collocation, could be particularly useful in the case of engineering analysis under uncertainty. Indeed the capability to obtain an error estimate on the prediction of the statistics of the objective functions, the relative use of a fewer number of samples with respect to Monte Carlo and Latin Hypercube Sampling and the capability to handle unconverged simulations at a sample locations are desired proprieties in handling the problem of optimization under uncertainty. The use of uncertainty quantification in combination with optimization methodologies will be discussed in chapter 3, while in the next chapter we introduce the mathematical formulation of deterministic optimization and we quantify the effect of uncertainties on the results.

Chapter 2

Optimization and Uncertainty

It is not the strongest of the species that survives, nor the most intelligent that survives. It is the one that is the most adaptable to change.

— Charles Darwin

Premature optimization is the root of all evil (or at least most of it) in programming.

— Donald Knuth

The concept of optimization is basic to much of what we do in our daily lives: a desire to do better or be the best in one field or another. In engineering we wish to produce the best possible result with the available resources. In a highly competitive modern world it is no longer sufficient to design a system whose performance of the required task is just satisfactory. It is essential to design the best system. Thus in designing new products in any field: aerospace, automotive, chemical, electrical, biomedical, agricultural, etc, we

must use design tools which provide the desired results in a timely and economical fashion. Numerical optimization is one of the tools at our disposal. Optimization is a very general automated design technique. In studying this technique it is important to distinguish between analysis and design. Analysis is the process of determining the response of the specified system to the certain combination of input parameters. For example, calculation stresses in the structure as a result of certain loads. Design on the other hand, means the process of defining a system. For example, designing a structure would mean selecting specific dimensions and location of the structural members that will allow the structure to withstand the specified load. Much of the design task in engineering is quantifiable, and so we are able to use computers to analyze alternative designs rapidly. The purpose of numerical optimization is to aid us in rationally searching among alternative designs for the best design to meet our needs. The alternative designs of the same system differ from each other because some parameters of the system are not the same. The parameters that could be changed in the system while searching for the best design are called design variables. Although we may not always think of it this way, design process may be defined as the process of finding the minimum or maximum of some characteristic, which may be called the objective function. For the design to be acceptable it must also satisfy certain requirements. These requirements are called design constraints. Optimization automatically changes the design variables to help us find the minimum or maximum of the objective function, while satisfying all the required design constraints. The desire to account for realistic and naturally varying

operating conditions leads to the formulation of the optimization problems in the presence of uncertainty. Engineering examples range from the inclusion of manufacturing tolerances in turbo-machinery design to the design of wind turbines under stochastically defined wind scenarios, from combustion stability control in the presence of fuel impurities to optimization of the operations of green buildings under variable occupancy and external conditions, and many others.

This chapter starts with a review of the mathematical formulation of the deterministic optimization problem (i.e. the objectives are deterministic), hence the focus is given to genetic algorithms in order to define the operators that will be generalized in chapter 4 to extend their usage to the framework of optimization under uncertainty.

A multi-objective application is then considered in a problem of the Formula 1 industry: the optimization of a F1 tire brake intake to maximize cooling efficiency and minimize aerodynamic resistance. It will be shown that considering the effect of uncertainties on the result of the deterministic optimization process could totally make meaningless a massive computational effort (e.g. large-scale, three-dimensional Reynolds-Averaged Navier-Stokes simulations on a high performance computing cluster) and that uncertainty need to be taken in account since the beginning of the design process.

2.1 The mathematical formulation

Problem formulation is normally the most difficult part of the process. It is the selection of design variables, constraints, objectives, and models of the

disciplines.

2.1.1 Design variables

A design variable is a specification that is controllable from the point of view of the designer. For instance, the thickness of a structural member can be considered a design variable. Another might be the choice of material. Design variables can be continuous (such as a wing span), discrete (such as the number of ribs in a wing), or boolean (such as whether to build a monoplane or a biplane). Design problems with continuous variables are normally solved more easily. Design variables are often bounded, that is, they often have maximum and minimum values. Depending on the solution method, these bounds can be treated as constraints or separately.

2.1.2 Constraints

A constraint is a condition that must be satisfied to make the design feasible. An example of a constraint in aircraft design is that the lift generated by a wing must be equal to the weight of the aircraft. In addition to physical laws, constraints can reflect resource limitations, user requirements, or bounds on the validity of the analysis models. Constraints can be used explicitly by the solution algorithm or can be incorporated into the objective using Lagrange multipliers or penalties.

2.1.3 Objectives

An objective is a numerical value that is to be maximized or minimized. For example, a designer may wish to maximize profit or minimize weight.

Many solution methods work only with single objectives. When using these methods, the designer normally weights the various objectives and sums them to form a single objective. Other methods allow multi-objective optimization, such as the calculation of a Pareto front, which will be defined in Section 2.3.

2.1.4 Models

The designer must also choose models to relate the constraints and the objectives to the design variables. These models are dependent on the discipline involved. They may be empirical models, such as a regression analysis of aircraft prices, theoretical models, such as from computational fluid dynamics, or reduced-order models of either of these. In choosing the models the designer must trade off fidelity with analysis time. The multidisciplinary nature of most design problems complicates model choice and implementation. Often several iterations are necessary between the disciplines in order to find the values of the objectives and constraints. As an example, the aerodynamic loads on a wing affect the structural deformation of the wing. The structural deformation in turn changes the shape of the wing and the aerodynamic loads. Therefore, in analyzing a wing, the aerodynamic and structural analyses must be run a number of times in turn until the loads and deformation converge.

2.1.5 Standard form

Once the design variables, constraints, objectives, and the relationships between them have been chosen, the problem can be expressed in a standard

form. Let's consider an objective function $f(z)$ where $z \in Z$ represents a design variable. A minimization problem is formulated in general as:

$$\begin{cases} f(\bar{z}) \leq f(z) & \forall z \in Z \\ \text{s.to:} & g(z) \leq 0, \quad h(z) = 0 \end{cases} \quad (2.1)$$

where $g(z)$ is a vector of inequality constraints, $h(z)$ is a vector of equality constraints.

2.2 Problem solution methods

The previous problem is normally solved using appropriate techniques from the field of optimization. These include gradient-based algorithms, population-based algorithms, or others. Very simple problems can sometimes be expressed linearly; in that case the techniques of linear programming are applicable. Most of the optimization techniques require large numbers of evaluations of the objectives and the constraints. The disciplinary models are often very complex and can take significant amounts of time for a single evaluation. The solution can therefore be extremely time-consuming. Many of the optimization techniques are adaptable to parallel computing. Much current research is focused on methods of decreasing the required time. Also, no existing solution method is guaranteed to find the global optimum of a general problem. Gradient-based methods find local optima with high reliability but are normally unable to escape a local optimum. Stochastic methods, like simulated annealing and genetic algorithms, will find a good solution with high probability, but very little can be said about the mathematical properties of the solution. It is not guaranteed to even be a local optimum. These

methods often find a different design each time they are run.

All the theoretical procedures presented in this thesis are general and not dependent on the problem solution but genetic algorithms will be used in the following to present the implementations of these concepts. Hence here it's presented a brief description of these class of algorithms.

2.2.1 Genetic algorithms

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness- a way to measure how they fit with the environment, represented by the objectives), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iter-

ation of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

2.3 The Pareto front

Having M objective functions, the notion of optimum changes, because in multi-objective problems, we are really trying to find good compromises (or trade-offs) rather than a single solution as in global optimization. The notion of optimum that is most commonly adopted is that originally proposed by Francis Ysidro Edgeworth in 1881. This notion was later generalized by Vilfredo Pareto (in 1896). Although some authors call Edgeworth-Pareto optimum to this notion, we will use the most commonly accepted term: Pareto optimum.

Definition 2.1

We say that a vector of decision variables $\bar{z} \in \tilde{Z}$ is Pareto optimal if there does not exist another $z \in \tilde{Z}$ such that $f_i(z) \leq f_i(\bar{z}), \forall i \in [1, \dots, M]$ and $f_i(z) < f_i(\bar{z})$ for at least the one i -th objective.

Here \tilde{Z} denotes the feasible region of the problem (i.e., where the constraints are satisfied). In words, this definition says that \bar{z} is Pareto optimal if there exists no feasible vector of decision variables $z \in \tilde{Z}$ which would decrease some objective without causing a simultaneous increase in at least

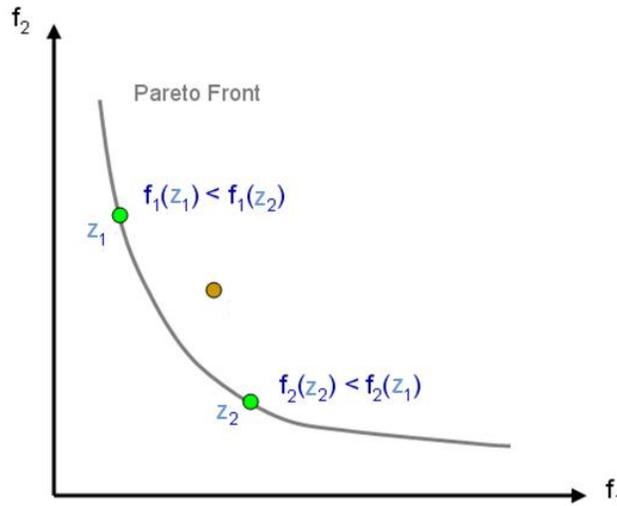


Figure 2.1: Pareto Front and non-domination illustration

one other objective. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the Pareto optimal set. The vectors \bar{z} corresponding to the solutions included in the Pareto optimal set are called non-dominated. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front, see Figure 2.1.

2.4 The NSGA-II algorithm

NSGA is a popular non-domination based genetic algorithm for multi-objective optimization. It is a very effective algorithm but has been generally criticized for its computational complexity, lack of elitism and for choosing the optimal parameter value for sharing parameter σ_{share} . A modified version, NSGA-II [63] was developed, which has a better sorting algorithm ,

incorporates elitism and no sharing parameter needs to be chosen a priori. The population is initialized based on the problem range and constraints, if any. Once initialized, the population is sorted based on non-domination and collected into a set of fronts: the first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and so on. To each individual is assigned a rank (fitness) value based on front in which it belong to. Individuals in the first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. In addition to fitness value a second parameter called crowding distance is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbors. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected if the rank is lesser than the other or if crowding distance is greater than the other. The population with the current population and current offsprings is sorted again based on non-domination and only the best N individuals are selected, where N is the population size. The selection is based on rank and the on crowding distance on the last front. The Non-dominated Sorted Genetic Algorithm (**NSGA-II**) is organized in several steps that consist of:

1. **Initialization of the population** The population is initialized based on the problem range and constraints if any. Usually the initial population is provided with a random seeding in the subset of the design space, Z ;

2. **Non-Dominated sort.** The initialized population is sorted based on non-domination: an individual is said to dominate another if the objective functions of it is no worse than the other and at least in one of its objective functions it is better than the other. This algorithm is better than the original NSGA since it utilize the information about the set that an individual dominate and number of individuals that dominate the individual. Once the non-dominated sort is complete the crowding distance is assigned. Since the individuals are selected based on rank and crowding distance all the individuals in the population are assigned a crowding distance value. Crowding distance is assigned front wise and comparing the crowding distance between two individuals in different front is meaning less. The basic idea behind the crowing distance is finding the euclidian distance between each individual in a front based on their M objectives in the M dimensional hyper space. The individuals in the boundary are always selected since they have infinite distance assignment. Further details can be found in [63];

Start of the evolution process (the following actions are performed in each generation)

3. **Selection of the parents.** Parents are selected for reproduction to generate offspring. The NSGA-II uses a binary tournament selection based on the crowded-comparison operator [63]. Tournament selection is carried out until the pool (i.e. the number of parents to be selected) size is filled,

4. **Crossover and mutation.** The selected population generates offsprings from crossover and mutation operators (e.g. Simulated Binary Crossover (SBX) and Polynomial mutation);
5. **Recombination and Selection.** Recombination and Selection. The offspring population is combined with the current generation population and selection is performed to set the individuals of the next generation. Since all the previous and current best individuals are added in the population, elitism is ensured. Population is now sorted based on non-domination. The new generation is filled by each front subsequently until the population size exceeds the current population size. If by adding all the individuals in front F_j the population exceeds N then individuals in front F_j are selected based on their crowding distance in the descending order until the population size is N ;

End of the evolution process

6. **Post-processing of the Pareto Front**

2.5 Case study: Optimization of a F1 wheel assembly

Formula 1 engineers are interested in primarily three factors related to tire aerodynamics i) overall tire lift and drag ii) cooling performance of the brakes and iii) how the tire airflow affects downstream components (wake characteristics). All three factors are tightly coupled which makes design quite complicated, especially when uncertainty in the flexible tire walls and

upstream conditions can negatively effect the car performance.

Figure 2.2 shows the wake sensitivity caused by flow traveling through the tire hub and exiting from the outboard side of the tire. If the flow of air is not allowed to pass through the tire hub (the top left and bottom left images in Figure 2.2), there is no mass efflux from the outboard side of the tire and the wake is quite symmetric about the wheel centerline. The wake is dominated by a counter-rotating vortex pair and both the inboard (left) and outboard (right) vortex are of similar size. Alternatively, if the flow of air is allowed to pass through the tire hub the inboard (left) vortex becomes larger than the outboard (right) vortex causing wake asymmetry (the top right and bottom right images in Figure 2.2).

The results of the single parameter perturbations indicated previously show the mass flow rate through the brake duct and tire drag force are more sensitive to the brake duct width than the brake duct height or length (in the range of deformation between $\pm 1\text{cm}$). The physical explanation of this result becomes evident when visualizing iso-contours of turbulent kinetic energy around the tire. Figure 2.2 shows the difference between a low width configuration (top) and high width configuration (bottom). The larger width of the brake duct causes a larger separation region immediately behind the brake duct in addition to higher turbulence levels in the shear layer immediately behind the inboard back edge of the tire.

In the next we analyze a nontrivial multi-objective problem in which it is not possible to find a unique solution that simultaneously optimizes each objective: when attempting to improve an objective further, other objec-

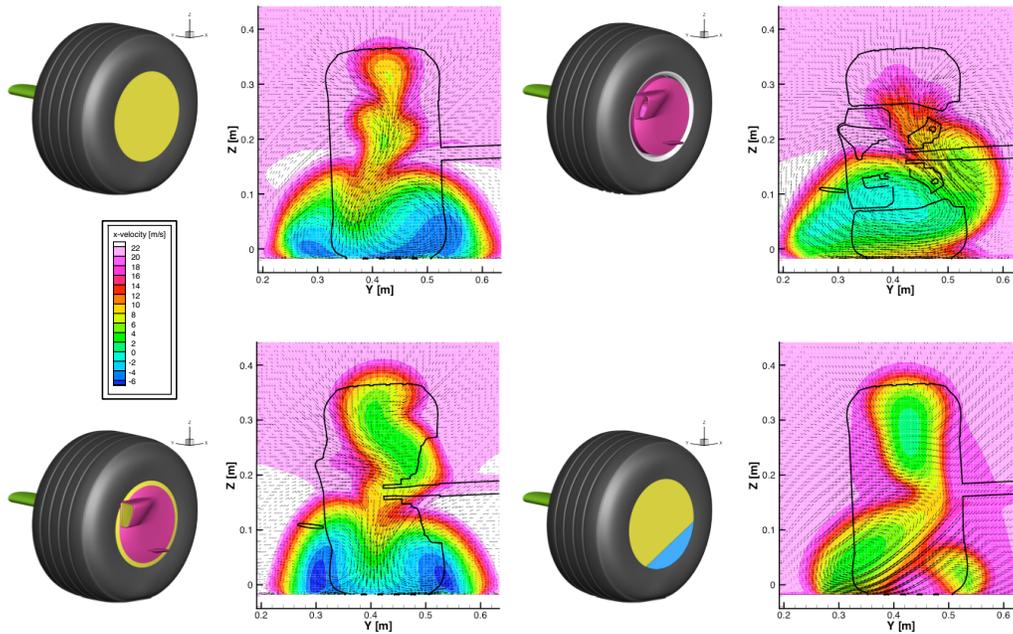


Figure 2.2: Wake sensitivity (shown by streamwise x-velocity contours for a plane located 1.12 wheel diameters downstream from the center of the tire) for a simplified tire with wheel fairings (top left), baseline F1 tire (top right), baseline F1 tire with blocked hub passages (bottom left), and simplified tire with artificial mass efflux from blue segment (bottom right)

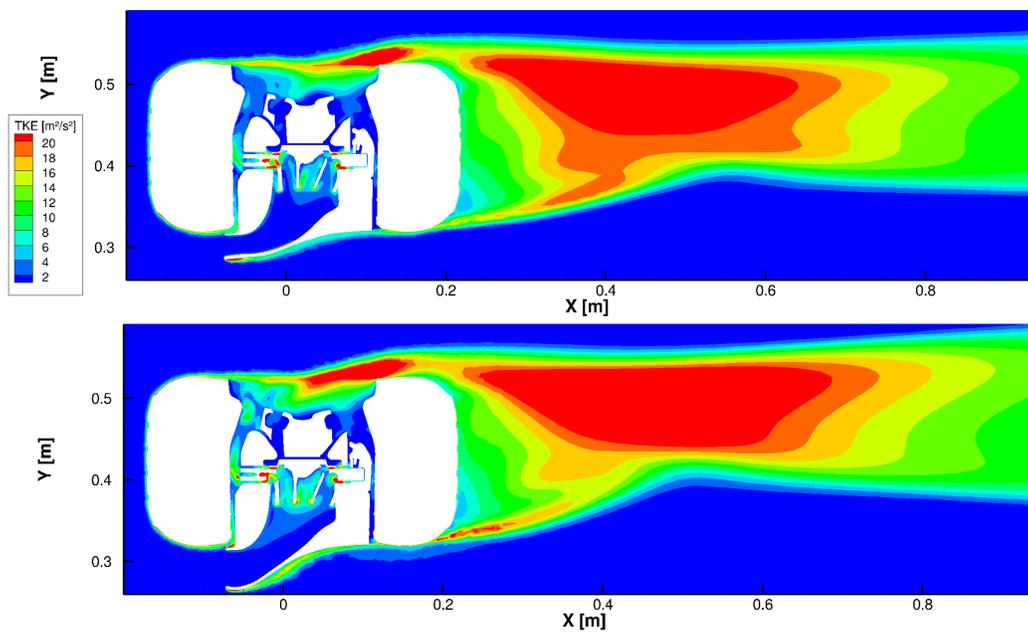


Figure 2.3: Turbulent kinetic energy contours for the minimum drag configuration (top) and maximum cooling configuration (bottom)

tives suffer as a result. A tentative solution is called non-dominated, Pareto optimal, or Pareto efficient if an improvement in one objective requires a degradation of another. We use the NSGA-II algorithm [63, 64] to obtain the non-dominated solutions, therefore we analyze the more interesting solutions on the deterministic Pareto set in presence of uncertainty. This is done in order to prove the importance of taking in account the variability of several input conditions in the design process.

2.5.1 The physics

Waschle [66] performed laser Doppler anemometry measurements in the wake of a stationary and rotating Formula 1 tire and compared the data to different numerical codes. In his experiments he was able to show reversed flow regions in the near wake, but the two main counter rotating vortices (CVP) near the ground were poorly captured due to low resolution. Mears [67] conducted a very elaborate study on stationary and rotating wheels with spokes. He used wheel pressure measurements, PIV, and steady RANS to show the near field as well as far field CVP structure. The results of this work highlights the need to investigate transient methodologies (both experimental and computational) in order to truly understand the intrinsically unsteady wake. RANS has been the most widely used computational method to compute the flow field around the tire. Skea [68] compared non-linear and standard turbulence models as well as various differencing schemes, and showed that the influence of the different models and schemes was significant. Recently, other approaches have become popular due to the well known issues

of RANS modeling. Waschle showed that improved predictions were possible using the Lattice-Boltzmann approach compared to RANS. McManus et al. [69] performed a computational study of both stationary and rotating wheels using the unsteady RANS (URANS) method and compared these to the Fackrell and Harvey [70] flow measurements. They showed good agreement with the experiments and gave a general schematic of the flow, including details of coherent structures that were not shown previously using RANS. Many computational studies have compared pressure contours around the surface of the tire using various turbulence models, but wake velocities and structures are more sensitive to turbulence treatment than forces on the tire (as shown in the results section). Axerio-Cilies [71, 65] shew that the wake behind a rotating simplified isolated tire is primarily dominated by a counter rotating vortex pair (CVP). The CVP is created by the downwash region behind the tire. The downwash is created by the side flow of the tire transferring energy to the top aft flow of the tire, causing the entrainment of flow from the top of the tire downwards. This downwards movement creates a downwash region which fuels the formation of the counterrotating ground vortex pair (CVP).

2.5.2 The cost

For such a study, there are approximately 400 simulations to perform per optimization cycle (i.e. generation). When the results of those 400 simulations are analyzed, an additional list of 400 simulations, each with a unique range of input parameters, are generated for the next generation in the optimization process. The values of the the input parameters for the

next generation are not known a priori. The optimization procedure needs to account for uncertainties arising from variable inflow conditions as well as variability in the flexible tire geometry. This complex baseline geometry consists of 30 million mesh cells. In order to generate an optimal design under uncertainty the mesh is deformed locally, creating 5000 unique simulations to perform. Each simulation (or realization) will be run on our in-house cluster using 2400 cores; the full design process should take approximately 2 weeks to complete.

2.5.3 The objectives

In this section the shape of a F1 brake duct is optimized, taking into account the geometrical uncertainties associated with the rotating rubber tire and uncertain inflow conditions. The objectives are to minimize the tire drag [N] while maximizing the captured mass flow (kg/s) needed to cool the brake assembly. A computational mesh consisting of 30 million elements is considered for a fully detailed 3D wheel model (Figure 2.5). Each simulation is based on a parallel CFD solver but the presence of geometrical uncertainties require the use of a local mesh deformation software.

2.5.4 Optimization Variables

A local mesh morphing software, Sculptor, was used to deform the baseline brake duct originally provided by Toyota Formula 1 (Figure 2.4). Specific control volumes were used to deform the brake duct in three dimensions, namely i) width of opening ii) height of opening and iii) protrusion length.

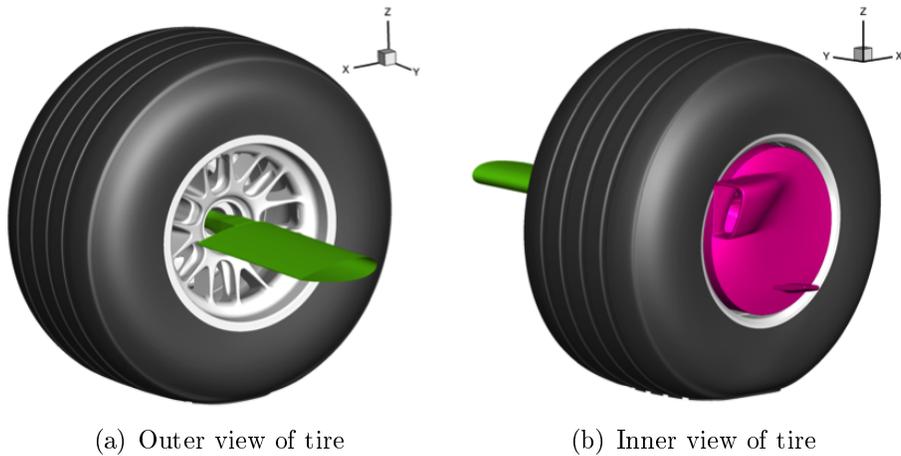


Figure 2.4: Front right tire of the Formula 1 race car used in this study showing green airfoil strut used to secure tire to the experimental wind tunnel facility and the outer brake duct (magenta) used to cool the brake assembly

Each design variable was allowed to change by $\pm 1\text{cm}$.

2.5.5 Uncertain Variables

Multiple uncertain variables were tested to determine their sensitivity to output quantities of interest using a DOE (design of experiments) approach. Some of the uncertain variables were based on the inflow conditions (i.e. yaw angle, turbulent intensity, turbulent length scale) while others were based on geometric characteristics of the tire (i.e. contact patch details, tire bulge radius, camber angle). Figure 2.7 shows 9 geometric modifications that were performed. Each subfigure shows the minimum, baseline F1 tire geometry, and maximum deformation for each uncertain variable.

From the results of purely a one-dimensional perturbation analysis the turbulence length scale (on the order of $0\text{m} \sim 2\text{m}$) results in less than a 0.1%

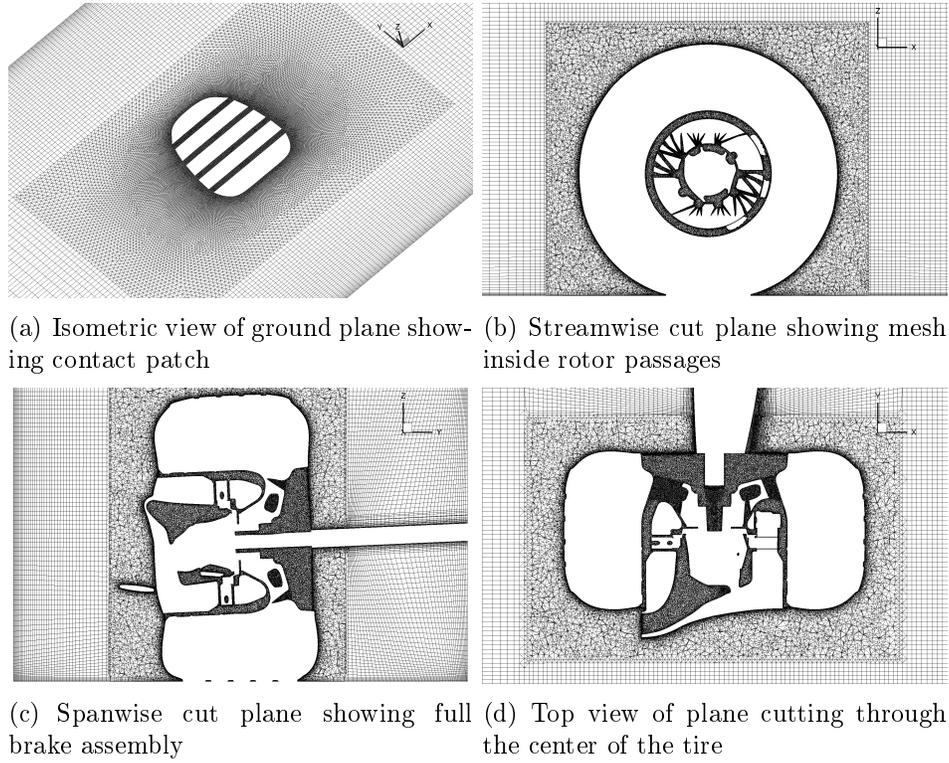


Figure 2.5: Four different views showing the Formula 1 tire mesh

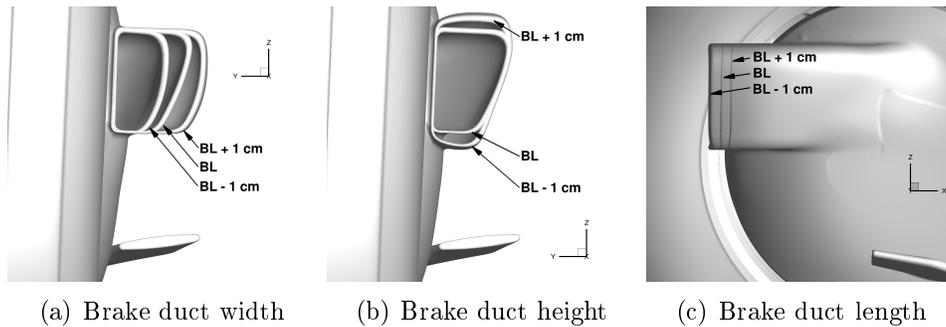


Figure 2.6: Brake duct optimization variables

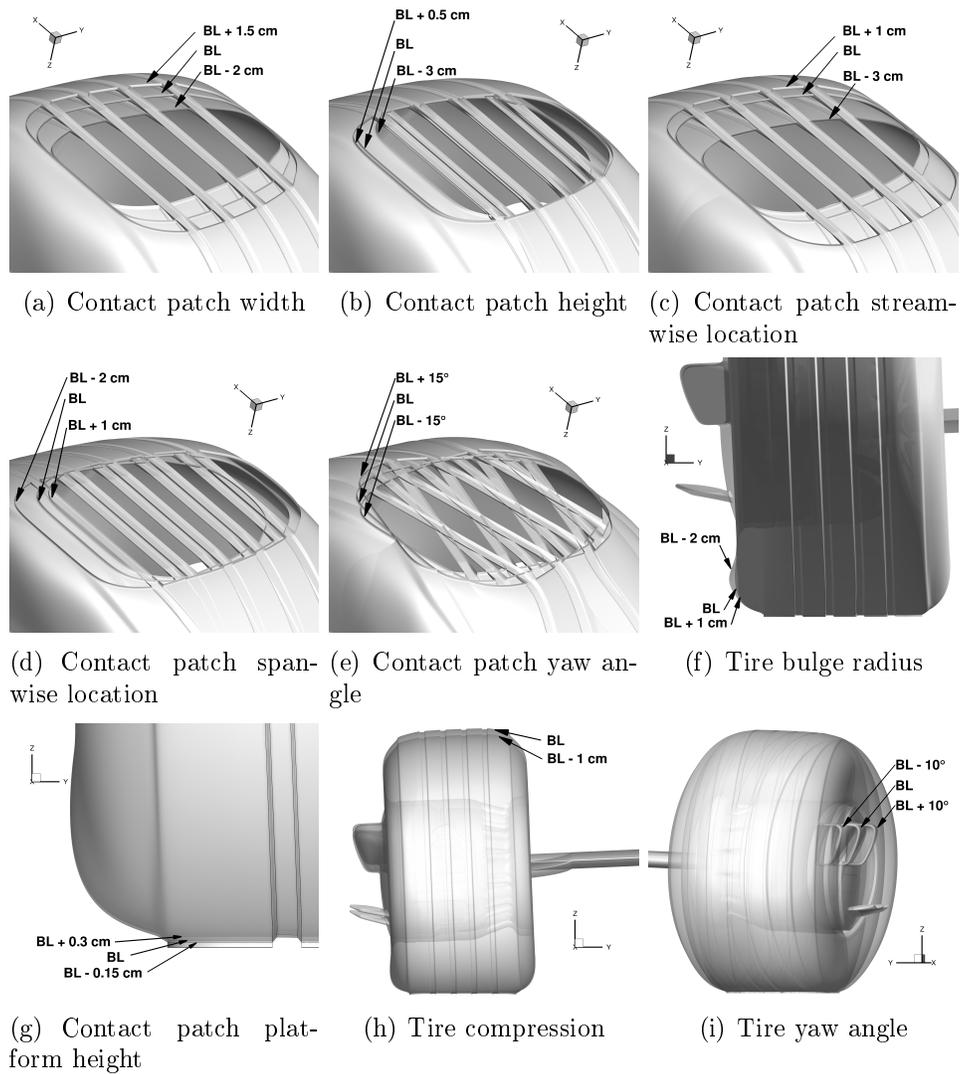


Figure 2.7: Subset of uncertain variables tested for sensitivity in output quantities of interest

difference in both the mass flow rate through the brake duct and overall drag on the tire. Conversely, both the mass flow rate and tire drag are very sensitive to the turbulence intensity. The mass flow rate decreased by 7.8% compared to the baseline (less cooling) with 40% turbulence intensity, and the tire drag increased by 7.2% with 40% turbulence intensity. This analysis confirms that the car performance decreases with ‘dirty’ air compared to ‘clean’ air. The sensitivity of the output quantities of interest caused by the tire yaw angle is shown in the first row of Table 2.1. The remaining rows in Table 2.1 show the sensitivity of mass flow rate and drag force to geometric characteristics, specifically contact patch, tire bulge radius, tire compression, and brake duct dimensions.

In the end, the three most sensitive uncertain variables, namely the tire contact patch width, tire yaw angle, and turbulence intensity were selected for the optimization under uncertainty study. The tire contact patch width was able to expand and contract up to 1cm, the tire yaw angle varied between $\pm 3^\circ$, and the turbulence intensity varied between 0% \sim 5%.

2.5.6 Deterministic Pareto Front

The Pareto frontier showing the optimal brake duct designs under no uncertainty are shown in Figure 2.8. Ten generations, which equates to 450 simulations, were needed to eventually construct the Pareto frontier. Further details about the optimization strategy can be found in Table 2.2. This table reports the settings of the NSGA-II algorithm adopted to drive the main phases of the genetic algorithm: selection (e.g. mating pool, parent

CHAPTER 2. OPTIMIZATION AND UNCERTAINTY

Table 2.1: Mass flow rate into the brake duct and drag force on the tire sensitivity for 9 uncertain variables and 3 design variables

	Deformation	Mass Flow Rate Change	Drag Force Change
Contact Patch Width [cm]	-2	-0.65%	3.65%
	-1	-0.34%	1.87%
	1	0.39%	-2.10%
	1.5	0.72%	-4.24%
Contact Patch Height [cm]	-3	-0.28%	-1.79%
	-2	-0.21%	-1.11%
	-1	-0.12%	-0.47%
	0.5	0.08%	0.18%
Contact Patch Streamwise Location [cm]	-3	0.78%	-4.73%
	-2	0.40%	-1.92%
	1	-0.13%	0.57%
Contact Patch Spanwise Location [cm]	-2	0.22%	-1.34%
	-1	0.04%	-0.20%
	1	0.12%	-0.41%
Contact Patch Yaw Angle [°]	-15	0.66%	-6.08%
	-10	0.31%	-2.50%
	10	-0.02%	0.02%
	15	0.24%	-1.99%
Tire Bulge Radius [cm]	-2	0.08%	0.38%
	-1.5	0.06%	0.39%
	1	0.01%	-0.51%
Contact Patch Platform Height [cm]	-0.15	0.49%	-3.28%
	-0.13	0.46%	-3.12%
	-0.11	0.41%	-2.72%
	-0.09	0.35%	-2.44%
	-0.07	0.29%	-1.90%
	-0.05	0.22%	-1.31%
	-0.03	0.17%	-0.98%
0.3	-0.21%	0.88%	
Tire Compression [cm]	-1	-2.06%	-6.44%
Tire Yaw Angle [°]	-10	1.93%	-1.03%
	10	-4.48%	6.12%
Brake Duct Width [cm]	-1	4.14%	1.46%
	1	-13.66%	-0.43%
Brake Duct Height [cm]	-1	-5.32%	0.98%
	1	3.33%	0.12%
Brake Duct Length [cm]	-1	-1.83%	-0.11%
	1	-2.85%	0.13%

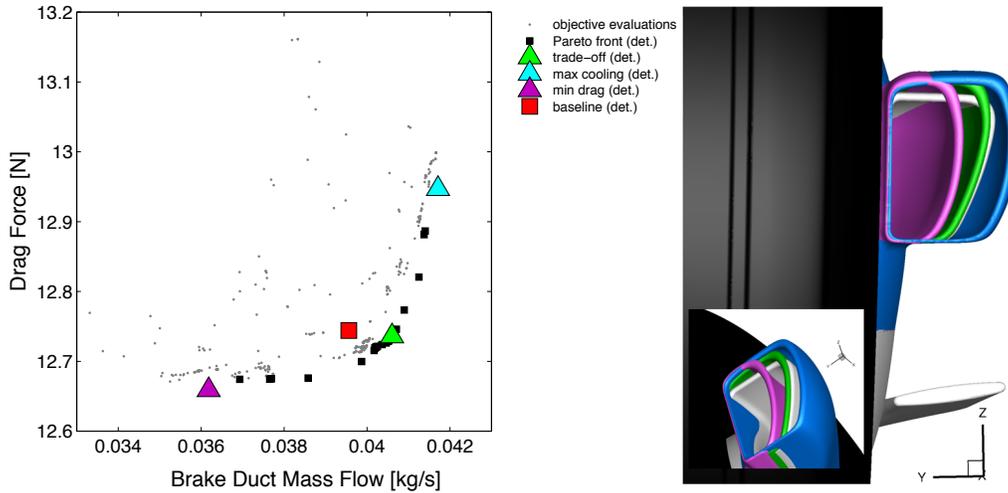


Figure 2.8: Deterministic Pareto front (left); the green, blue, magenta, and gray brake ducts in the subfigure on the right correspond to the trade-off, max cooling, minimum drag, and baseline configurations respectively

sorting)[63] and reproduction (e.g. crossover and mutation)[63, 64]. Leland (see chapter 5) was used to handle the job scheduling and management and as a result the time required to complete the 450 simulations was 2 days compared to about 4 days without using Leland, which requires submitting jobs manually to the job queuing system using a constant number of processors. Among the Pareto set (see Figure 2.8), the design that achieves the highest mass flow rate is shown in blue and the design that achieves the lowest overall drag on the tire is shown in magenta. The green design is labeled as the trade-off design since this design tries to achieve the highest mass flow through the inlet of the brake duct while minimizing the total drag on the tire. The baseline geometry, reported in red, was shown not to be on the Pareto front in the deterministic setting.

Table 2.2: Multi-objective optimization strategy

Parameter	Value
Population size [-]	50
Crossover fraction [-]	0.90
Mutation fraction [-]	0.10
Parent sorting	Tournament between couples
Mating Pool [%]	50
Crossover mode	Simulated Binary Crossover (SBX)
Generations [-]	10

2.5.7 Analysis under Uncertainty of Pareto Front

In the previous results once the tire configuration and other input conditions are specified, the solution is uniquely determined without vagueness. On the other hand, when uncertainties are present, the results have to be expressed in a non-deterministic fashion either probabilistically or as ranges of possible outcomes. The approach we followed here using the SSC is strictly non-intrusive, in the sense that the existing tools are used without modifications, but the solution - or more precisely, their probability distributions - are constructed performing an ensemble of deterministic analyses. Further details about the uncertainty quantification strategy can be found in Table 2.3.

The variability of the four geometries described above (namely trade-off, highest mass flow, lowest drag, and baseline) as a result of the uncertainties in the the tire yaw angle, turbulence intensity, and contact patch width are shown in Figure 2.9. The variability of the minimum drag design is highest shown by the spread of magenta dots, followed by the maximum mass flow

Table 2.3: Uncertainty quantification strategy

Parameter	Value
UQ algorithm[-]	SSC
Maximum number of SSC samples[-]	30
Convergence Threshold on refinement [-]	$1e^{-3}$
Number of Monte Carlo samples [-]	$1e^3$
Polynomial order of interpolation [-]	automatic up to 6

design shown by blue dots, trade-off design shown by green dots and baseline design shown by red dots. The colored dots in this figure represent the mean probabilistic values and the black lines represent ± 1 standard deviation of the probabilistic distribution. It is evident in this figure that the optimal designs, on average, move away from Pareto frontier, decreasing the overall performance of the race car.

A similar conclusion can be drawn by looking at the probability density of the drag force and the brake mass flow (Figure 2.10). The latter shows a large excursion of both the position of the peak and the support, while the former is only marginally affected. This directional sensitivity under uncertainty with respect to brake duct mass flow might suggest that only the brake duct mass flow maximization could be treated as a probabilistic objective, while the drag reduction optimization can be handled using conventional (deterministic) optimization.

The variability of the four geometries described above (namely trade-off, highest mass flow, lowest drag, and baseline) to uncertainties in the the tire yaw angle and contact patch width are shown in Figure 2.9. The variability of the minimum drag design is highest shown by the spread of magenta dots,

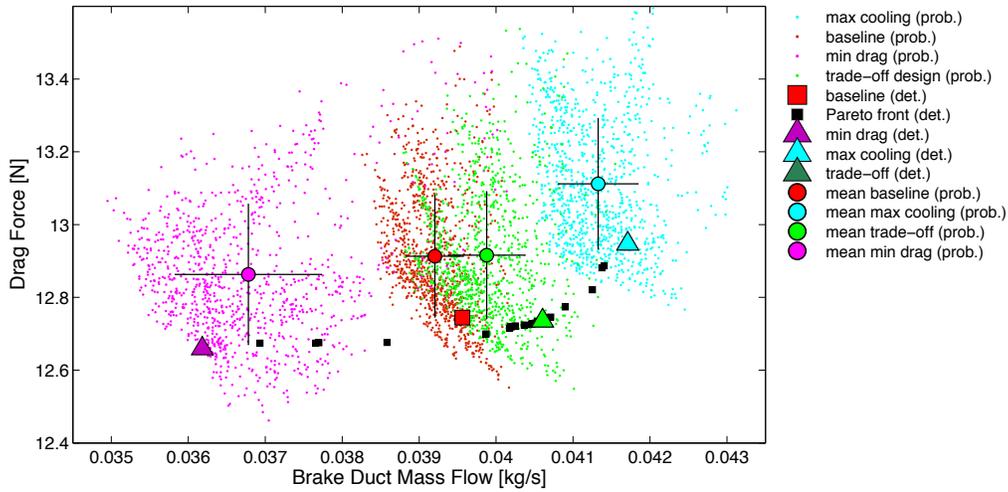


Figure 2.9: Pareto frontier for F1 wheel assembly showing the variability of the minimum drag (magenta), baseline (red), trade-off (green), and maximum cooling (blue) designs to uncertainty in the inflow conditions and flexible tire geometry.

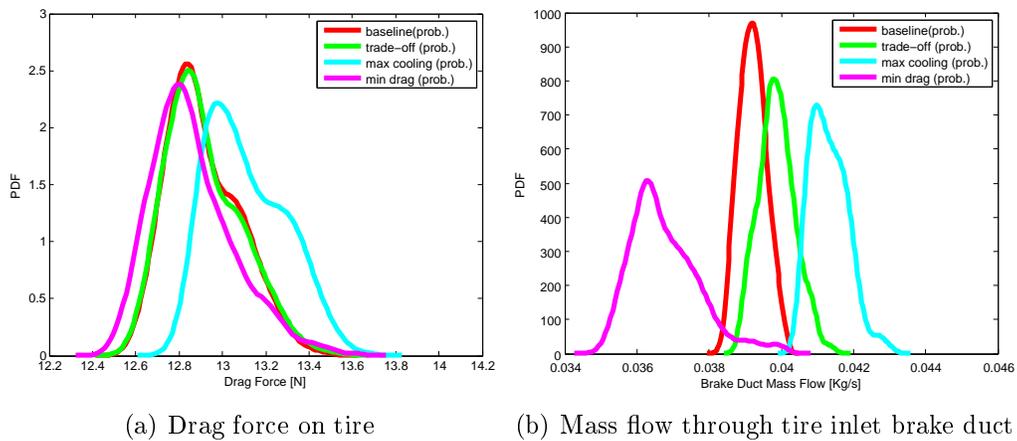


Figure 2.10: PDF's of the output quantities of interest used for this study

followed by the maximum mass flow design shown by blue dots, trade-off design shown by green dots and baseline design shown by red dots. It is evident in this figure that the optimal designs, on average, move away from Pareto frontier, decreasing the overall performance of the race car.

2.6 Lessons learned

The analyzed F1 optimization problem proved that the optimization process cannot be decoupled from the uncertainty quantification process since the solutions identified above move away from the deterministic Pareto in presence of uncertainty. The cost of performing complex CFD simulation on high performance computing is totally vanished when the natural variability of several parameters is considered. This is the main point on which we focus in the rest of this thesis: the coupling between uncertainty quantification and optimization. Since all the information that can be obtained from the propagation of the uncertainties can be summarized in the PDFs (or the CDFs) of the objectives, the use of these informations to build a tight coupling with optimization will be explored and novel algorithms presented with respect to the classical frameworks. Indeed we move apart from the classical approaches for optimization under uncertainty that adopt the same deterministic procedures and algorithms while using the statistical moments of the targets as deterministic objectives. In the next chapter we consider the single-objective formulation of the problem of optimization under uncertainty, while in chapter 4 we will extend the proposed methodologies to the multi-objective framework.

Chapter 3

Single-Objective Optimization under Uncertainty

Optimization under uncertainty is an extension of conventional optimization procedures and aims at taking in account uncertainty in the design procedure. In this chapter it will be introduced a new framework characterized by the use of all the possible informations in the probabilistic domain, namely the Cumulative Distribution Function (CDF), which represents the identity card of a design analyzed under uncertainty. Due to this peculiarity this approach sets itself apart from the conventional methods which rely on the use of few statistical moments as deterministic attributes in replacing the objectives of the optimization process. Additionally the use of an area metric leads to a multi-objective methodology which allows an a posteriori selection of the candidate design based on risk/opportunity criteria of the designer.

3.1 The mathematical formulation

The concept of robust optimization is intuitively connected to the idea that in the presence of (input) uncertainty the optimal design should be relatively insensitive (small output uncertainty). We will review the commonly used robustness principles before introducing a novel, and more general, definition.

Let's consider an objective function $f(z, \xi)$ where $z \in Z$ represents a design variable and $\xi \in \Omega$ a random variable (which can be either a design variable or another parameter in the problem). A minimization problem is formulated in general as: find $\bar{z} \in Z$ such that

$$f(\bar{z}, \xi) \leq f(z, \xi) \quad \forall z \in Z, \quad \forall \xi \in \Omega \quad (3.1)$$

In a probabilistic framework for uncertainty analysis (such as the one introduced earlier) the problem is that $f(z, \xi)$ is a random quantity induced by ξ . It is possible to introduce an operator Φ , applied to $f(z, \xi)$ in order to obtain a deterministic attribute of it, reducing the problem to finding $\bar{z} \in Z$ such that

$$\Phi(f(\bar{z}, \xi)) \leq \Phi(f(z, \xi)) \quad \forall z \in Z \quad (3.2)$$

Different definition for Φ might be used, for example $\Phi(f(z, \xi))$ are the statistical moments of f . The simplest choice is obviously the expected value of f (referred to as **Mean Value Optimization** [72]):

$$\Phi(f(z, \xi)) = \int_{\Omega} f(z, \xi) \Psi_{\xi} d\xi = \mu(z) \quad (3.3)$$

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

where Ψ_ξ is the probability density function of ξ . The main advantage of this approach is that the mean is the fastest converging statistical moment, meaning that relatively few samples (e.g. Latin Hypercube Sampling, Stochastic Collocation, etc.) are required to obtain an accurate prediction. On the other hand, the mean is not a sufficient representation of a complete probability distribution. Indeed other (higher order) moments can be used (**Mean Value Penalty Optimization** [73, 74]):

$$\Phi(f(z, \xi)) = w_1 \mu(z) + \left(\sum_{k=2}^N w_k m^k(f(z, \xi)) \right)^{1/2} \quad (3.4)$$

where w_1, \dots, w_N are (tunable) weights, N is the maximum order of statistical moments considered and $m^k(f(z, \xi))$ is the k-th order moment of $f(z, \xi)$

$$m^k(f(z, \xi)) = \int_{\Omega} (f(z, \xi) - \mu(z))^k \Psi_\xi d\xi \quad (3.5)$$

which leads to (for $w_1 = w_2 = 1$ and $N = 2$)

$$\Phi(f(z, \xi)) = \mu(z) + \sigma(z) \quad (3.6)$$

where $\sigma^2(z)$ is the variance of $f(z, \xi)$. In this case the optimization under uncertainty seeks to minimize the mean plus standard deviation, giving a formal and mathematically sound construction for the idea of insensitive design. The main advantage of this methodology is that additional (but still not sufficient) informations about the probability distribution could be used to shape an appropriate objective function while, on the other hand, the weights required could be not known precisely.

A different, extremely conservative, approach is the **Minimax Principle** [72] where the problem is finding $\bar{z} \in Z$ such that

$$\sup_{\Omega} f(\bar{z}, \xi) \leq \sup_{\Omega} f(z, \xi) \quad \forall z \in Z \quad (3.7)$$

This approach seeks to protect against the worst-case scenario and the main drawback is the related excess of conservatism.

Despite these fundamental approaches, the widely adopted methods in practical applications come from the concepts of i) constrained optimization and ii) multi-objective optimization. In the first case, interpreted as a **Constrained Optimization** [74], the problem is to find $\bar{z} \in Z$ such that

$$\begin{cases} \mu(\bar{z}) \leq \mu(z) \quad \forall z \in Z \\ \text{s.to:} & m^k(f(z, \xi)) \leq C_k \quad \forall k \in 2, N \end{cases} \quad (3.8)$$

where C_k is a constraint on the order k central moment of $f(z, \xi)$. Again for $N=2$ this procedure reduces in finding $\bar{z} \in Z$ such that

$$\begin{cases} \mu(\bar{z}) \leq \mu(z) \quad \forall z \in Z \\ \text{s.to:} & \sigma^2(z) \leq \sigma^{2*} \end{cases} \quad (3.9)$$

where σ^* is the maximum value allowed for the variance. The main advantage of this methodology is to inject additional informations about the probability distribution trough a constraint based on higher order statistical moments. It easy to notice, as drawbacks, that i) the constraint could not be feasible or ii) the constraint may lead to skip a design superior to the others for any realization of the random variable. As final strategy it is possible to formulate the problem as a **Multi-objective Approach**[73] in the form

$$\begin{cases} \min_Z \mu(z) \\ \min_Z m^k(f(z, \xi)) \quad \forall k \in 2, N \end{cases} \quad (3.10)$$

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

which for $N=2$ becomes

$$\begin{cases} \min_Z \mu(z) \\ \min_Z \sigma^2(z) \end{cases} \quad (3.11)$$

The main advantage of this approach is an a posteriori decision making of the candidate design, indeed different statistical moments are used as independent trade-off objectives. In this case a challenge is posed by the increase in dimensionality since an original M multi-objective problem turns into a $M \times N$ multi-objective problem, where N is the number of the statistical moments used in this formulation. Even in this case few statistical moments could not be representative of the full probability distribution. Additionally this methodology requires decision making criteria on the Pareto Front to select the candidate design. The presence of correlations between statistical moments could violate the underlying hypotheses of independence of the objectives.

Method	Features	Advantages	Drawbacks
Mean Value Approach	The mean is used as objective to reduce the dependency on ξ	The mean is the faster converging moment and few samples are required	Often the mean is not a representative summary of the distribution
Mean Value Penalty Optimization	The mean is penalized by a weighted sum of the higher order moments	It allows to take in account a penalty based on higher order moments	The weights are unknown and few statistical moments could be not enough to represent the distribution
Minimax Principle	The limit superior (worst case due by uncertainty) assumed as objective	Protection against worst case scenario	Excess of conservatism
Constrained Optimization	Same as Mean Value Approach but with a precise specification of constraints on higher order moments	Excludes designs which don't respect constraints	The constraint might not be feasible and few statistical moments could be not enough to represent the distribution
Multi-objective Optimization	Different moments are used as different objectives	A posteriori choice of the candidate design	Increase of dimensionality and not well-suited in presence of correlation between moments

Figure 3.1: Literature review of Optimization under Uncertainty

3.2 A novel CDF based approach

In the approaches described above the objective function is transformed into a different form in order to eliminate its random character (reduce the dependency on ξ). One of the consequences is that typically only the mean and the variance of the objective function are considered in the process, thus potentially eliminating important characteristics of its probabilistic description (only in special, simple cases, the mean and the variance completely describe the overall distribution, e.g Gaussian). The goal of the approach proposed here [82] is to i) avoid any assumptions on the type of the objective function i.e. Gaussianity and ii) avoid increasing the dimensionality of problem (as in the multiobjective robust optimization).

Given the expectations of the designer a desired template probabilistic distribution, $\tau(z)$, can be defined a priori in order to guide the optimization process. Every actual design, being subject to uncertainty, will be represented by a distribution with a non-zero support. The main concept we introduce is to identify a measure of the difference between the CDFs of the template and any other design to direct the optimization process. Many possible measures of distance between distributions can be considered; here we use the area metric [23], also referred to as the Minkowski L_1 metric; the area between the CDF of the template and the CDF of the candidate design, $F(z)$, is the measure of the mismatch between them and gives information about the robustness of the candidate design (RI - Robustness Index)

$$RI(z) = \int_{\Omega} |F(z) - \tau(z)| d\xi \quad (3.12)$$

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

The robust optimum can be formulated as to find $\bar{z} \in Z$ such that

$$RI(\bar{z}) \leq RI(z) \quad \forall z \in Z \quad (3.13)$$

which for $RI(z)=0$ returns the ideal design corresponding to the assumed template.

The RI generalizes the deterministic comparison of scalar values that have no uncertainty or the difference between statistical moments (as used in many robust optimization algorithms); if both the template and the candidate design are deterministic, the CDFs are Dirac functions and in this case the area is equal to the difference between two scalar values. The RI will not to be overly sensitive to minor discrepancies in the distribution tails (assuming the area is finite), because it reflects the full distribution of the scalar point values. In particular, it is clearly not merely a measure of the difference in the means and/or variances, but takes into account the overall difference between distributions. Another useful property of this measure, is that its units are identical to the objective function units[23].

In Figure 3.2 a uniform distribution is assumed as template of the robust optimization process. Using the area metric it is possible to state that the candidate Beta distribution is closer to the template than the candidate LogN distribution.

The previous definition can be used in solving the problem in Equation 3.1. Let's assume a reference solution, namely the RAO (Reference Absolute Optimum), which we define as the inferior limit of the objective function and which corresponds to an ideal deterministic optimum

$$RAO = \inf_{Z \times \Omega} f(z, \xi) \quad (3.14)$$

The ideal solution of a problem of optimization under uncertainty is a design that ensures the lowest possible value of the objective function (the RAO) with no sensitivity on the uncertain quantities.

In terms of the Cumulative Density Function (CDF) the RAO is represented by a vertical line and assumed to be a reference for the proposed method, hence the Equation 3.12 becomes

$$RI(z) = \int_{\Omega} |F(z) - \delta_z(RAO)| d\xi = \mu(f(z, \xi)) - RAO \quad (3.15)$$

where δ_z is the Dirac delta function.

Using the concept of area measure, the robust optimum can be formulated to finding $\bar{z} \in Z$ such that

$$RI(\bar{z}) \leq RI(z) \quad \forall z \in Z \quad (3.16)$$

which for $RI(z)=0$ returns the ideal design corresponding to the RAO.

3.3 Global and local robustness

In a large number of applications the minimization problem in (3.1) could have no global solution $\forall \xi \in \Omega$. If we restrict the global problem to a subspace $\bar{\Omega} \in \Omega$ it could be possible to find a solution for the local problem: find $\bar{z} \in Z$ such that

$$f(\bar{z}, \xi) \leq f(z, \xi) \quad \forall z \in Z, \quad \forall \xi \in \bar{\Omega} \quad (3.17)$$

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

Due to this consideration it is possible to generalize the concept of Robustness Index introduced in Section 3.2. We introduce a local Robustness Index, $RI_a^b(z)$

$$RI_a^b(z) = \int_a^b |F(z) - \delta_z(RAO)| d\xi \quad (3.18)$$

where $a, b \in [0, 1]$ are two probability thresholds on the Cumulative Distribution Function. Considering N non overlapping partitions $\bar{\Omega}_i$ of Ω such that $\sum_1^N \bar{\Omega}_i = \Omega$, due to the additive propriety of the area measure

$$\sum_1^N RI_{\bar{\Omega}_i}(z) = RI(z) \quad (3.19)$$

Considering the partition of $\Omega = [0, 1]$ in $\Omega_1 = [0, 0.2]$, $\Omega_2 = [0.2, 0.8]$ and $\Omega_3 = [0.8, 1]$ of Figure 3.4, due to the previous considerations

$$RI_0^{0.2}(z) + RI_{0.2}^{0.8}(z) + RI_{0.8}^1(z) = RI(z) \quad (3.20)$$

Considering the first partition of Ω it is possible to deduct that

$$RI_0^{0.2}(U(0, 1)) < RI_0^{0.2}(\beta(2, 2)) < RI_0^{0.2}(\beta(4, 4)) \quad (3.21)$$

A $RI_0^{p_1}(z)$ based optimization could be interpreted as an optimization strategy in which we distinguish the candidate designs based on the favorable tail of the distribution: in this example the uniform distribution provides the greatest probability to manifest a behavior closer to the RAO in Ω_1 . Indeed in several contexts ensuring at least a certain probability of excellent designs could be the design goal of a technological process. Similarly it is possible to follow a strategy that includes the minimization of the probability of extremely poor realizations - a typical goal in reliability-based optimization.

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

Considering the second partition of Ω it is possible to deduct that

$$RI_{0.2}^{0.8}(U(0, 1)) = RI_{0.2}^{0.8}(\beta(2, 2)) = RI_{0.2}^{0.8}(\beta(4, 4)) \quad (3.22)$$

A $RI_{p_1}^{p_2}(z)$ based optimization could be interpreted as percentile based strategy in which the tails of the distributions are not considered due to difficulties in convergence or admissible tolerances : in this example all the distributions have the same behavior in Ω_2 due to symmetry.

Considering the last partition of Ω it is possible to deduct that

$$RI_{0.8}^1(U(0, 1)) > RI_{0.8}^1(\beta(2, 2)) > RI_{0.8}^1(\beta(4, 4)) \quad (3.23)$$

A $RI_{p_2}^1(z)$ based optimization could be interpreted as a generalization of the **Minimax Principle** in which the worst case scenario is replaced by the unfavorable tail of the distribution: in this example the $\beta(4, 4)$ distribution present the greatest probability to manifest a behavior closer to the RAO in Ω_3 .

The additive propriety of the RI leads to the multi-objective approach that will be introduced in the next Section.

3.4 A CDF partition based approach

In nontrivial multi-objective problems it is not possible to find a unique solution that simultaneously optimizes each objective: when attempting to improve an objective further, other objectives suffer as a result. A tentative solution is called non-dominated, Pareto optimal, or Pareto efficient if an improvement in one objective requires a degradation of another. Given the

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

N non overlapping partitions $\bar{\Omega}_i$ of Ω such that $\sum_1^N \bar{\Omega}_i = \Omega$, it is possible to define in mathematical terms the multi-objective problem as

$$\min_Z [RI_{\bar{\Omega}_1}(z), RI_{\bar{\Omega}_2}(z), \dots, RI_{\bar{\Omega}_N}(z)]^T \quad (3.24)$$

where $[RI_{\bar{\Omega}_1}(z), RI_{\bar{\Omega}_2}(z), \dots, RI_{\bar{\Omega}_N}(z)]^T$ is a vector of independent objectives.

Finding such non-dominated solutions, and quantifying the trade-offs in satisfying the different objectives, is the goal when setting up and solving the multi-objective optimization problem. Due to the additive property of the area metric it is interesting to notice an important consequence on the a posteriori choice of the candidate design from the Pareto Front: the Best Efficient Point on the Pareto Front, characterized by a minimum value of $\sum_1^N RI_{\bar{\Omega}_i}(z)$, corresponds to a single objective optimization in Ω .

3.5 Single-objective analytic test functions

In this section the effectiveness of the CDF partition based multi-objective approach introduced in Section 3.4 is verified on two analytic test cases, namely the Rosenbrock [76, 77] and the Goldstein-Pricek [78] functions. These functions are well-know test cases for deterministic optimization algorithms. In the follow we extend these deterministic problems to robust optimization test cases by considering a subset of the design variables as uncertainties characterized by assigned distributions of probability.

3.5.1 Rosenbrock function

The two-dimensional Rosenbrock function is a popular test problem for optimization algorithms due to the presence of several local minima. In Eldred [79] it was shown to be a challenging problem for certain UQ methods (especially local reliability methods), because a particular response level contour involves a highly nonlinear curve that may encircle the mean point (leading to multiple most probable points of failure in local reliability methods).

The function is a fourth-order polynomial

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad (3.25)$$

A three-dimensional plot of this function is shown in Figure 3.5. Taking x_1 to be a design variable with bounds $[-2, 2]$ and taking x_2 to be a standard normal random variable $\mathcal{N}(\mu = 0; \sigma = 0.2)$ it is possible to state the effectiveness of the multi-objective approach. Considering the three non overlapping partitions $\bar{\Omega}_1 = [0, 0.1]$, $\bar{\Omega}_2 = [0.1, 0.9]$ and $\bar{\Omega}_3 = [0.9, 1]$ in $\Omega = [0, 1]$ such that $\sum_1^N \bar{\Omega}_i = \Omega$, it is possible to define in mathematical terms the multi-objective problem as

$$\min_Z [RI_0^{0.1}(z), RI_0^{0.9}(z), \dots, RI_0^1(z)]^T \quad (3.26)$$

The NSGA-2 algorithm[63] has been used to obtain the non dominated solutions shown in Figure 3.7.a when $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$. Further details about the optimization strategy can be found in Table 3.1.

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

Table 3.1: Single-objective optimization strategy

Parameter	Value
Population size [-]	50
Crossover fraction [-]	0.90
Mutation fraction [-]	0.10
Parent sorting	Tournament between couples
Mating Pool [%]	50
Crossover mode	Simulated Binary Crossover (SBX)[64]
Generations [-]	1000
Uncertainty Quantification	Monte Carlo
Number of samples [-]	5×10^3

The projections of the Pareto Front on the three Cartesian planes have been reported in Figures 3.7.b-d.

The Pareto Front reveals that, due to the separation of the different objectives, different optimal solutions can be found in the restrictions of the probability space Ω . Among the non-dominated solutions the more interesting ones, corresponding to the minimum values assumed by the objectives have been marked on the Pareto Front. As mentioned in Section 3.4, the single-objective optimization strategy in Ω is obtained as a free result of the multi-objective optimization strategy and corresponds to the Best Efficient Point (BEP) [80] identified by the min RI solution.

The CDFs in all the partitions of Ω have been reported in Figures 3.8.a-d. It is evident the superiority of a candidate CDF in the corresponding probability restriction in which it was optimized. According to the optimization goal it will be possible to state a posteriori the optimal solution of the problem. In the case in which a 10% of events closer to RAO want to be

ensured by the technological process, the design corresponding to $\min RI_0^{0.1}$ would be the one preferred by the designer. In the case in which the priority is to reduce the unfavorable tail of the distribution corresponding to under performing designs the solution corresponding to $\min RI_{0.9}^1$ would be the one adopted by the designer. If the tails of the distributions are not important to the designer the solution corresponding to $\min RI_{0.1}^{0.9}$ would be the best trade-off solution to this purpose.

3.5.2 Goldstein-Price function

The Goldstein-Price function is a global optimization test function. It has only two independent variables and the following definition

$$f(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \cdot [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \quad (3.27)$$

A three-dimensional plot of this function is shown in Figure 3.6, where both x_1 and x_2 range in value in $[-2, 2]$.

The Goldstein-Price global minimum equal to $f(z, \xi) = 3$ is obtainable for $(x_1, x_2) = (0, -1)$. Taking x_1 to be a design variable with bounds $[-2, 2]$ and taking x_2 to be a standard normal random variable $\mathcal{N}(\mu = 0; \sigma = 0.2)$, it is possible to challenge the multi-objective approach in the same fashion we did for the Rosenbrock function. The non dominated solutions are shown in Figure 3.9.a. The projections of the Pareto Front on the three Cartesian planes have been reported in Figures 3.9.b-d. The structure of the Pareto Front is more complex with respect to the Rosenbrock test function and in

this example the result of a single objective optimization in Ω equals the solution of the problem restriction in Ω_3 as reported in Table 3.2.

The CDFs in all the partitions of Ω have been reported in Figures 3.10.a-d. It is interesting to notice that the candidate design corresponding to $RI_0^{0.1}$ highly over-perform all the others in Ω_1 but shows a very long under-performing tail. On the other end the design corresponding to $\min RI$ is characterized by the shortest under-performing tail (e.g. the worst performance is 4 times smaller than the previous design). In this example it is clearly evident that the multi-objective approach, giving the possibility of an 'a-posteriori' selection of the candidate design, allows to choose among very different solutions according to the designer's goal.

The relevant solutions corresponding to the previous analyzed problems of design under uncertainty have been summarized in Table 3.2.

Table 3.2: Relevant candidate designs

candidate design	Rosenbrock function	Goldstein Price's function
$\min RI_0^{0.1}$	$\bar{x}_1 = 0.5139$	$\bar{x}_1 = -0.6474$
$\min RI_{0.1}^{0.9}$	$\bar{x}_1 = 0.2130$	$\bar{x}_1 = 1.4548$
$\min RI_{0.9}^1$	$\bar{x}_1 = 0.0732$	$\bar{x}_1 = 1.1918$
$\min RI$	$\bar{x}_1 = 0.1798$	$\bar{x}_1 = 1.1918$

3.6 Lessons learned

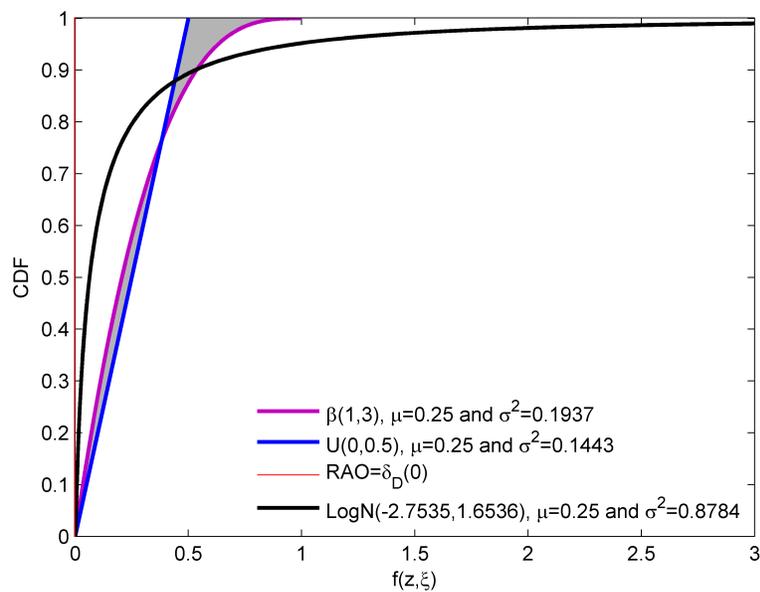
In this chapter a novel definition has been proposed to optimize in presence of uncertainty based on the use of a desired template distribution. The main advantage of this approach is the use of all possible information in the

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

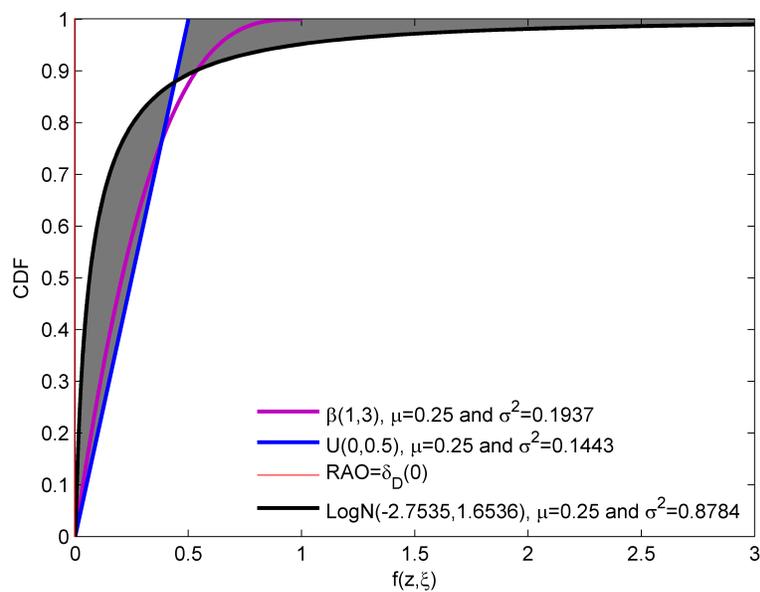
probabilistic domain, summarized in the CDFs of the designs. The capability to handle as single-objective problem with tools of the multi-objective framework (e.g. the NSGA-II algorithm) appeared very promising. Indeed the multi-objective approach was proven to offer an 'a-posteriori' selection of the candidate design based on the way in which the designer contemplate uncertainty in his vision of the optimization problem. In future works it will be assessed the robustness of this method when the CDF of the objective function is poorly resolved.

The concepts introduced here will be extended to multi-objective optimization in the next chapter.

CHAPTER 3. SINGLE-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

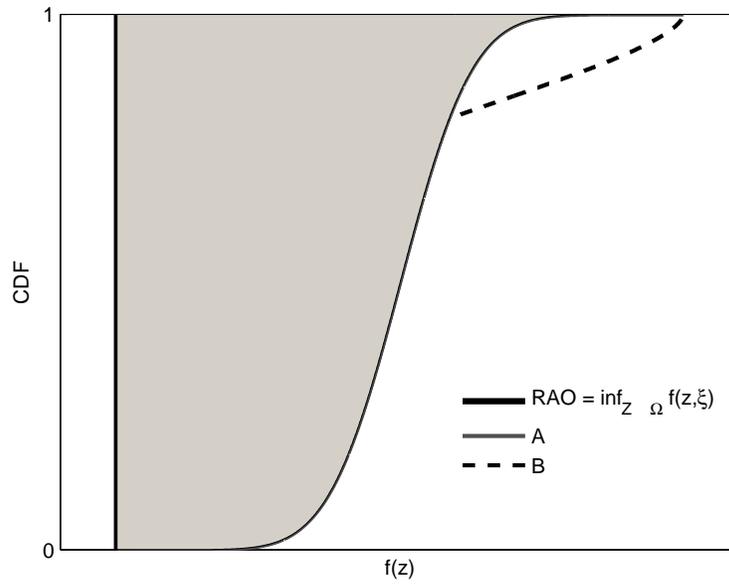


(a) The shaded area is the RI for Beta

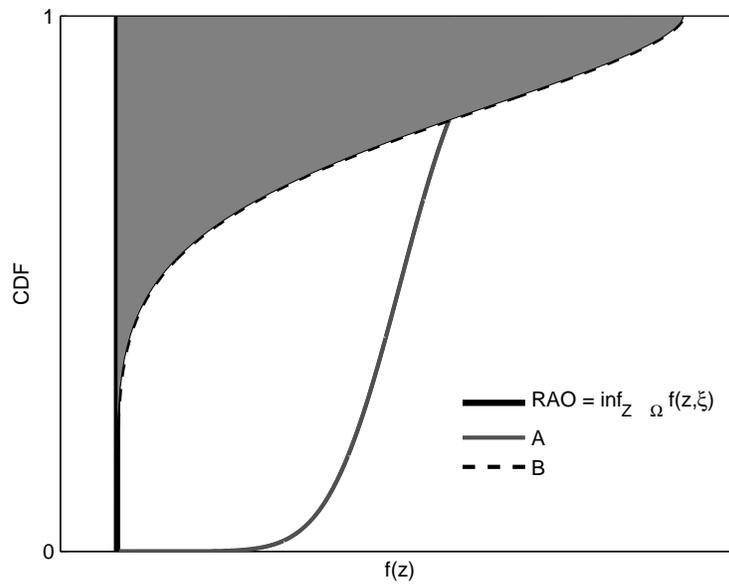


(b) The shaded area is the RI for LogN

Figure 3.2: Probabilistic distance from a Uniform template

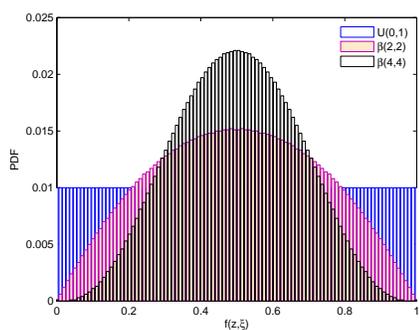


(a) The shaded area is the RI for candidate design A

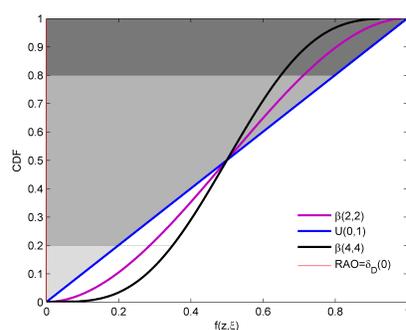


(b) The shaded area is the RI for candidate design B

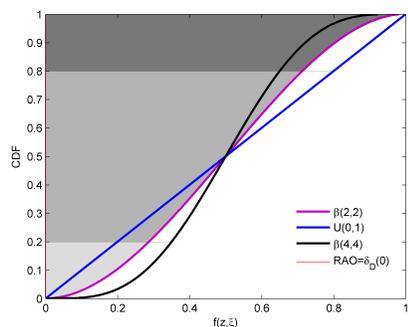
Figure 3.3: CDF based measure of robustness



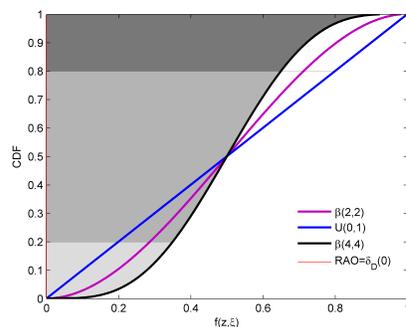
(a) Probability Density Functions



(b) The shaded area is the RI for $\beta(2,2)$



(c) The shaded area is the RI for $U(0,1)$



(d) The shaded area is the RI for $\beta(4,4)$

Figure 3.4: Additive propriety of the RI

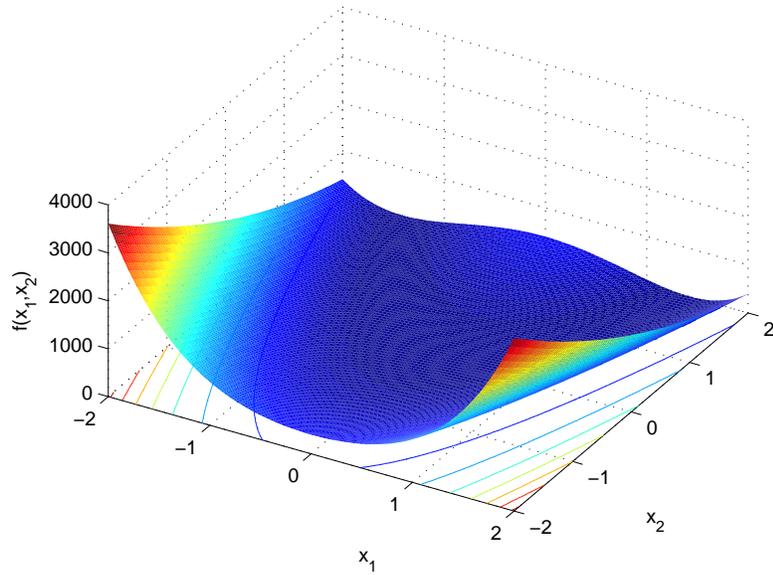


Figure 3.5: The Rosenbrock test function

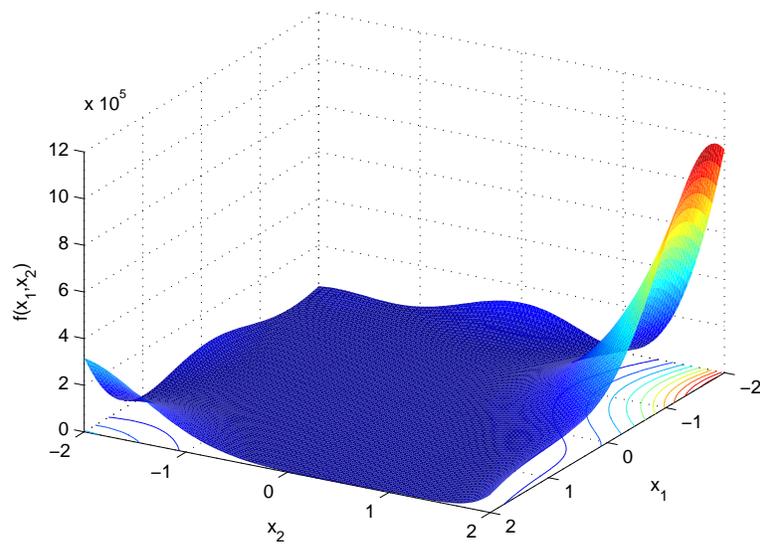


Figure 3.6: The Goldstein-Price test function

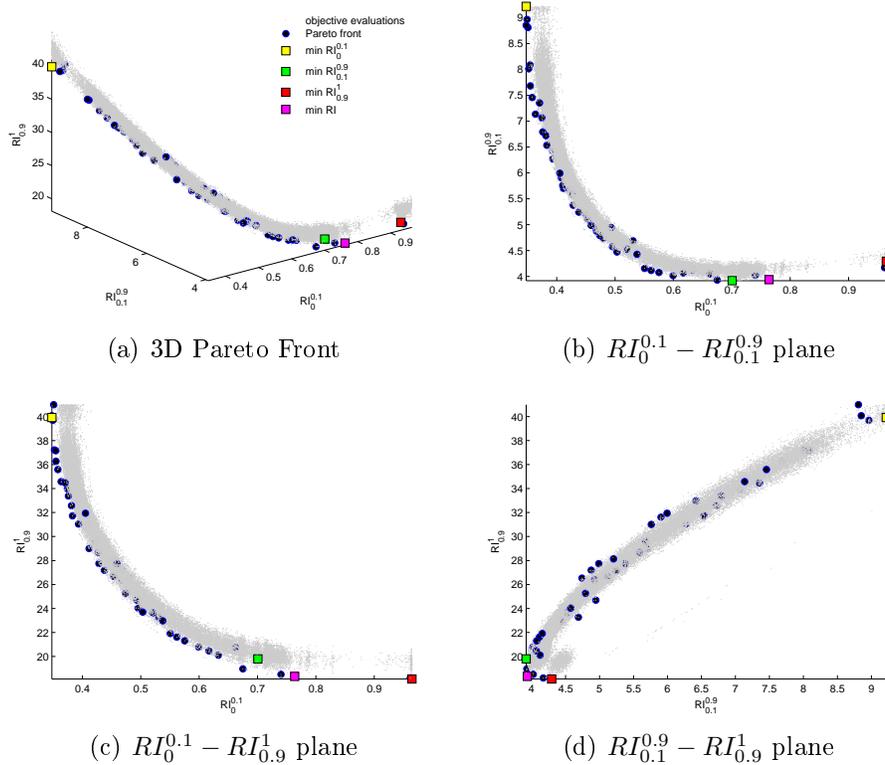


Figure 3.7: Rosenbrock function. Pareto Front for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$

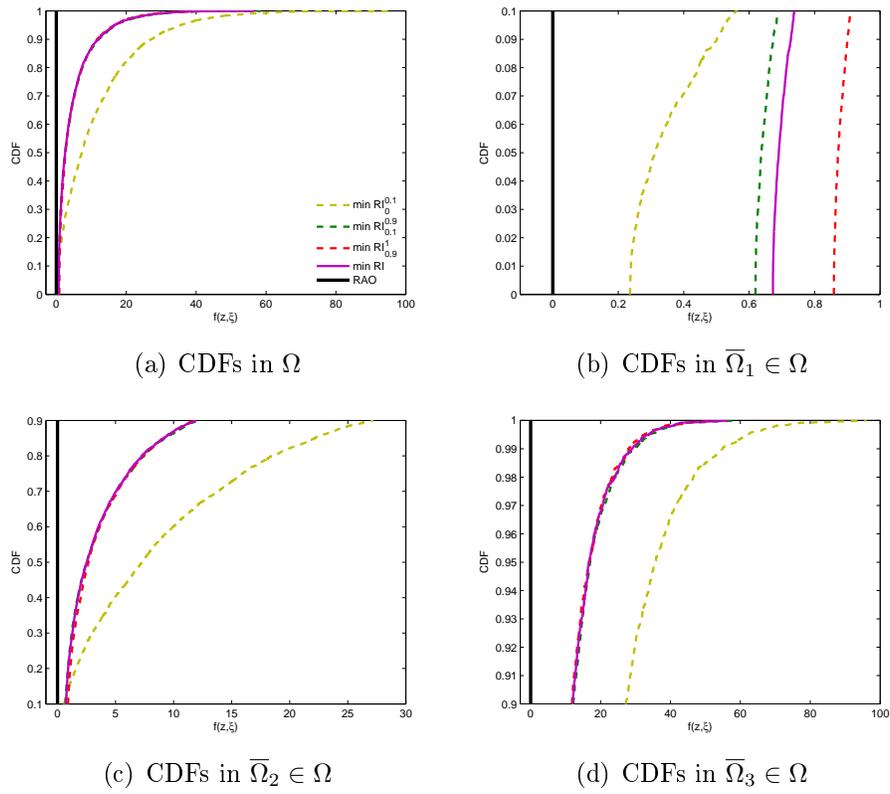


Figure 3.8: Rosenbrock function. CDFs for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$

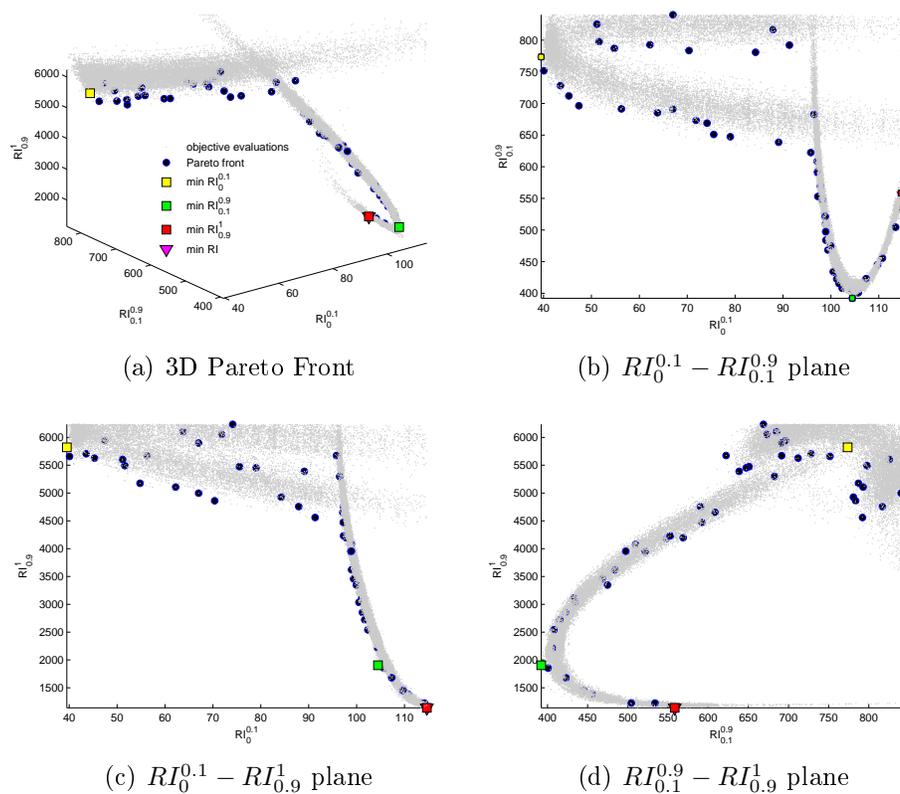


Figure 3.9: Goldstein-Price function. Pareto Front for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$

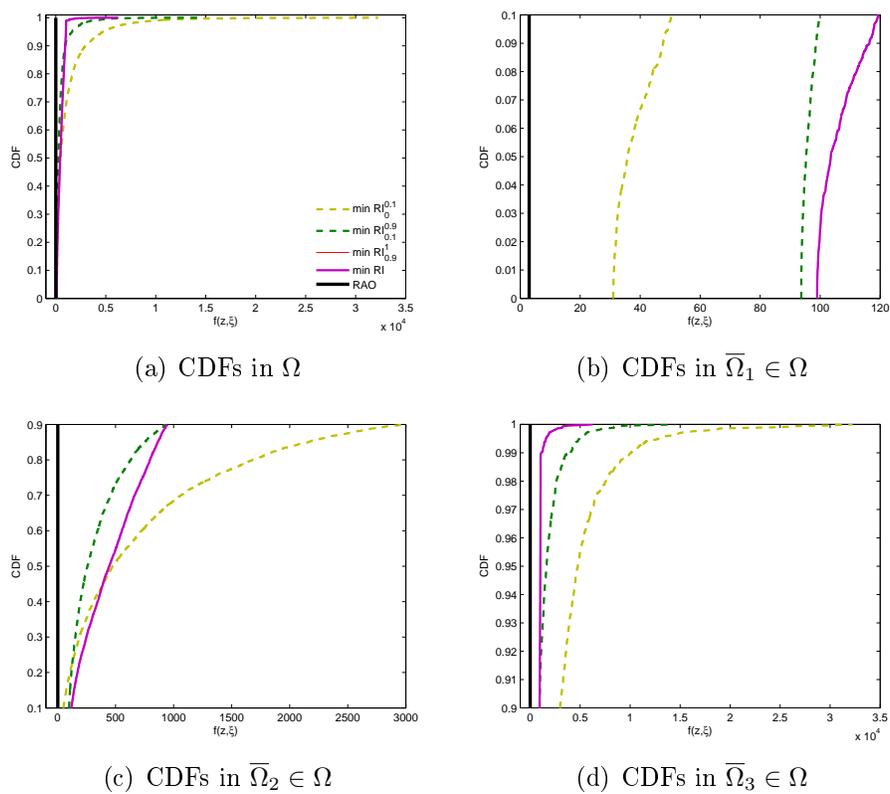


Figure 3.10: Goldstein-Price function. CDFs for $x_2 = \mathcal{N}(\mu = 0; \sigma = 0.2)$

Chapter 4

Multi-Objective Optimization under Uncertainty

The definitions introduced in the previous chapter for single-objective problems are now extended to multi-objective optimization problems. Furthermore the NSGA-II algorithm, described in chapter 2, is generalized to work in tight coupling with uncertainty quantification algorithms. A process of blade design in presence of insect contamination on the AOC 15/50 wind turbine is presented as an application of this multi-objective framework.

4.1 The mathematical formulation

Let's consider a set of objective functions $[f_1(z, \xi), f_2(z, \xi), \dots, f_M(z, \xi)]$ where $z \in Z$ represents a vector of design variables and $\xi \in \Omega$ a vector of random variables. In nontrivial multi-objective problems it is not possible to find a unique solution that simultaneously optimizes each objective: when attempting to improve an objective further, other objectives suffer as a result. A tentative solution is called non-dominated, Pareto optimal, or Pareto

efficient if an improvement in one objective requires a degradation of another. It is possible to define in mathematical terms the multi-objective problem as

$$\min_Z [f_1(z, \xi), f_2(z, \xi), \dots, f_M(z, \xi)]^T \quad (4.1)$$

where $[f_1(z, \xi), f_2(z, \xi), \dots, f_M(z, \xi)]^T$ is a vector of M independent objectives. Finding such non-dominated solutions, and quantifying the trade-offs in satisfying the different objectives, is the goal when setting up and solving the multi-objective optimization problem.

In a probabilistic framework for uncertainty analysis the problem is that each $f_i(z, \xi)$ is a random quantity induced by ξ . It is possible to introduce an operator Φ , applied to $f_i(z, \xi)$ in order to obtain a deterministic attribute of it, reducing the problem as

$$\min_Z [\Phi(f_1(z, \xi)), \Phi(f_2(z, \xi)), \dots, \Phi(f_M(z, \xi))]^T \quad (4.2)$$

Different definition for Φ might be used, for example $\Phi(f_i(z, \xi))$ can be the statistical moments of f_i . The simplest choice is obviously the expected value of f_i (referred to as **Mean Value Optimization** [72]):

$$\Phi(f_i(z, \xi)) = \mu_i(z) = \int_{\Omega} f_i(z, \xi) \Psi_{\xi} d\xi \quad (4.3)$$

where Ψ_{ξ} is the probability density function of ξ . Using this approach the optimization problem can be formulated as

$$\min_Z [\mu_1(z), \mu_2(z), \dots, \mu_M(z)]^T \quad (4.4)$$

and the resulting non-dominated solutions will be referred to as **Mean Value Pareto Front**.

The main advantage of this approach is that the mean is the fastest converging statistical moment, meaning that relatively few samples (e.g. Latin Hypercube Sampling, Stochastic Collocation, etc.) are required to obtain an accurate prediction. On the other hand, the mean is not a sufficient representation of a complete probability distribution.

Indeed other (higher order) moments can be used (**Mean Value Penalty Optimization** [73, 74]):

$$\Phi(f_i(z, \xi)) = w_1^i \mu_i(z) + \left(\sum_{k=2}^{N_M} w_k^i m^k(f_i(z, \xi)) \right)^{1/2} \quad (4.5)$$

where $w_1^i, \dots, w_{N_M}^i$ are (tunable) weights, N_M is the maximum order of statistical moments considered and $m^k(f_i(z, \xi))$ is the k-th order moment of $f_i(z, \xi)$

$$m^k(f_i(z, \xi)) = \int_{\Omega} (f_i(z, \xi) - \mu_i(z))^k \Psi_{\xi} d\xi \quad (4.6)$$

which leads to (for $w_1^i = w_2^i = 1$ and $N_M = 2$)

$$\Phi(f_i(z, \xi)) = \mu_i(z) + \sigma_i(z) \quad (4.7)$$

where $\sigma_i(z)$ is the standard deviation of $f_i(z, \xi)$. In this case the optimization under uncertainty seeks to minimize the mean plus standard deviation, giving a formal and mathematically sound construction for the idea of insensitive design.

Using this approach the optimization problem can be formulated as

$$\min_Z [\mu_1(z) + \sigma_1(z), \mu_2(z) + \sigma_2(z), \dots, \mu_M(z) + \sigma_M(z)]^T \quad (4.8)$$

and the resulting non-dominated solutions will be referred to as **Mean Value Penalty Pareto Front**.

The main advantage of this methodology is that additional (but still not sufficient) informations about the probability distribution could be used to shape an appropriate objective function while, on the other hand, the weights required could be not known precisely.

A different, extremely conservative, approach is the **Minimax Principle** [72]. Using this approach the optimization problem can be formulated as

$$\min_Z [\sup_{\Omega} f_1(z, \xi), \sup_{\Omega} f_2(z, \xi), \dots, \sup_{\Omega} f_M(z, \xi)]^T \quad (4.9)$$

and the resulting non-dominated solutions will be referred to as **Minimax Pareto Front**.

This approach seeks to protect against the worst-case scenario and the main drawback is the related excess of conservatism.

Despite these fundamental approaches a widely adopted method in practical applications come from the concepts of constrained optimization [74]. Using this approach the optimization problem can be formulated as

$$\begin{cases} \min_Z [\mu_1(z), \mu_2(z), \dots, \mu_N(z)]^T \\ \text{s.to: } m^k(f_i(z, \xi)) \leq C_k^i \quad \forall k \in [2, N_M], \forall i \in [1, M] \end{cases} \quad (4.10)$$

where C_k^i is a constraint on the order k central moment of $f_i(z, \xi)$. The resulting non-dominated solutions will be referred to as **Constrained Mean**

Value Pareto Front.

The main advantage of this methodology is to inject additional informations about the probability distribution through a constraint based on higher order statistical moments. It is easy to notice, as drawbacks, that i) the constraint could not be feasible or ii) the constraint may lead to skip a design superior to the others for any realization of the random variable.

While for a single objective problem it is widely adopted the **Multi-objective Approach** [73], where different statistical moments are used as independent trade-off objectives, in this case a challenge is posed by the increase in dimensionality since an original M multi-objective problem turns into a $M \times N_M$ multi-objective problem. Indeed a $M \times N_M$ dimensional Pareto Front is difficult to handle in practical applications.

A straightforward extension of the CDF partition based approach (see Section 3.4) is directly obtained considering a vectorial counterpart $\mathbf{\Phi}(\cdot) = (\Phi_1(\cdot), \dots, \Phi_{N_P}(\cdot))$ of the operator $\Phi(\cdot)$ where the N_P are non overlapping partitions used to evaluate the Robustness Index (see Section 3.3).

4.2 The generalization of the NSGA-II sorting operators

In this section we extend the concept of Robustness Index of Section 3.2 to multi-objective optimization problems using the concept of Pareto dominance, specifically in the context of Genetic Algorithms (GAs) [81]. In the NSGA-II algorithm [63] each individual in the population is assigned a numerical **rank** based on fitness, which is used together with a **crowding**

distance (e.g. a measure of how close an individual is to its neighbors) in the sorting procedure. Hence in this section we want to obtain probabilistic counterparts of these mechanisms that would generalize the GA for robust optimization while reverting to the original operators when the objective functions are deterministic [82].

4.2.1 A probabilistic definition of the rank

We recall that the RI is a measure of the difference between the CDFs of the desired target of the optimization under uncertainty process and any other design (see Section 3.2).

1. For each generation we assume the reference solution to be the one that dominates all others in the sense of Pareto (i.e. rank equal to 1) and assume its CDF to be deterministic.
2. Assuming an ensemble of realizations of each $f_i(z, \xi)$ obtained by sampling the uncertainties in Ω , for each design it is possible to rank the N individuals in a candidate population $z_1, z_2, \dots, z_j, \dots, z_N$ using the concept of Pareto dominance.
3. Repeating the same procedure for each realization of the uncertain variables, $\xi \in \Omega$, we can construct the CDF of the rank, $R(z)$, for each one of the candidate designs.
4. Using the RI is possible to obtain a measure of the probabilistic distance of $R(z)$ from the complete probabilistic domination in the sense of

Pareto as follows

$$\overline{RI}(z) = \int_0^1 |R(z) - \delta_z(1)| d\xi = \mu(R(z)) - 1$$

where $\delta_z(1)$ is the Dirac delta function centered on the complete probabilistic domination in the sense of Pareto.

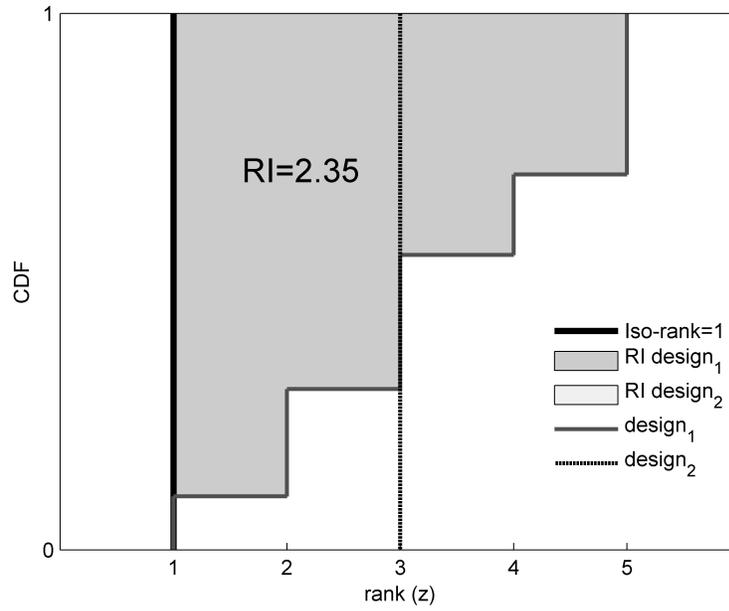
5. The previous measure can be used to rank the j -th candidate design (i.e. **probabilistic rank**)

$$j_{rank}^P = 1 + \overline{RI}_j$$

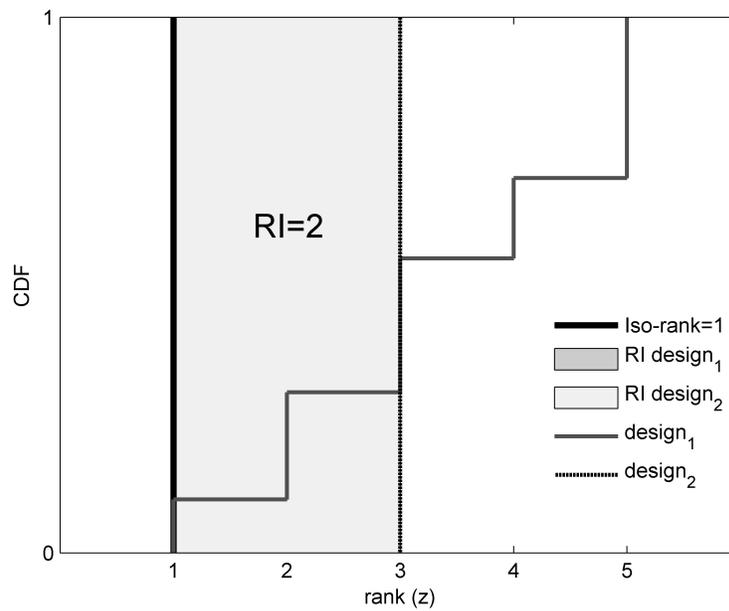
The probabilistic rank likely results in a real number if the distribution of $R(z)$ is not deterministic. The use of real numbers instead of integer numbers in the ranking procedure allows to inject additional information about the uncertainties inside the genetic algorithm, leading to a tight coupling with the uncertainty quantification process.

4.2.2 A probabilistic definition of the crowding distance

The basic idea behind the crowding distance is finding the euclidean distance between each individual in a front based on their M objectives in the M -dimensional hyper space. The crowding-distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each i -th objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other j -th intermediate solutions are assigned a distance value, $j_{distance}$, equal to the absolute normalized difference in the function values of two adjacent solutions



(a) The \overline{RI} of $design_1$ is 2.35



(b) The \overline{RI} of $design_2$ is 2

Figure 4.1: The use of Robustness Index as probabilistic rank

$$j_{distance} = \sum_{i=1}^M \frac{f_{j+1}^i - f_{j-1}^i}{f_{max}^i - f_{min}^i} \quad (4.11)$$

where f_{j+1}^i and f_{j-1}^i are the i -th objectives for the solutions which respectively follow and precede the j -th after the sorting procedure and f_{max}^i and f_{min}^i are respectively the maximum and the minimum values of the i -th objective in the current population.

While in the deterministic case a solution is represented by a point in the M dimensional hyper space of the objectives, in the probabilistic case a solution is fully characterized by the probability distribution of the objectives and represented by a cloud of realizations. Hence we can define a **probabilistic crowding distance** as follows

1. For each i -th objective we sort the population $z_1, z_2, \dots, z_j, \dots, z_N$ according to the mean of $f_i(z, \xi)$
2. We compute the distance

$$j_{distance}^P \quad i = \frac{\mu(f_{j+1}^i) - \mu(f_{j-1}^i)}{\mu(f_{max}^i) - \mu(f_{min}^i)}$$

3. Repeating the same procedure for each objective we can estimate the probabilistic crowding distance

$$j_{distance}^P = \sum_{i=1}^M j_{distance}^P \quad i = \sum_{i=1}^M \frac{\mu(f_{j+1}^i) - \mu(f_{j-1}^i)}{\mu(f_{max}^i) - \mu(f_{min}^i)}$$

4.2.3 A probabilistic Crowded-Comparison Operator

Using the previously described mechanisms we want to generalize the crowded-comparison operator (\prec_n) which guides the selection process at the

various stages of the algorithm toward a Pareto optimal front. Assume that every j -th individual in the population has three attributes:

1. integer part of the probabilistic rank, $(\lfloor j_{rank}^P \rfloor)$;
2. non integer part of the probabilistic rank, $(j_{rank}^P - \lfloor j_{rank}^P \rfloor)$;
3. probabilistic crowding distance, $(j_{distance}^P)$.

It was noticed that selecting two out of these three informations to create a partial order (crowded-comparison-operator, (\prec_n)) similar to the one of the NSGA-II algorithm, would have led to the following problems

1. using the integer part of the probabilistic rank as first selection criterion and the non integer part of the probabilistic rank as second selection criterion leads to a clustering of solution around the one characterized by the highest value of the probabilistic rank (i.e. absence of diversity preservation)
2. using the integer part of the probabilistic rank as first selection criterion and the non integer part of the probabilistic rank as second selection criterion leads to a partial order that does not turn back to its deterministic counterpart when designs are deterministic (i.e. absence of consistency)
3. using the integer part of the probabilistic rank as first selection criterion and the probabilistic crowding distance as second selection criteria ensures an uniform spread of solutions but leads to a loss of part of the

informations obtained from uncertainty quantification (i.e. absence of tight coupling between optimization and uncertainty quantification)

In the same fashion Deb used the rank to distinguish the solutions among fronts, we use here the integer part of the probabilistic rank to this purpose while we use the non integer part of the probabilistic rank and the probabilistic crowding distance to define a secondary classification criterion. In a particular front, the **probabilistic sub-rank** is obtained considering the non integer part of the probabilistic rank and the probabilistic crowding distance as two trade-off requirements and performing a sorting based on the Pareto dominance among them.

Considering the \bar{j} -th and the $\bar{\bar{j}}$ -th individuals, we now define a probabilistic partial order (\prec_n^P) as

$$\begin{aligned} \bar{j} \prec_n^P \bar{\bar{j}} \quad & \text{if} \quad \left(\lfloor \bar{j}_{rank}^P \rfloor < \lfloor \bar{\bar{j}}_{rank}^P \rfloor \right) \\ & \text{or} \quad \left(\lfloor \bar{j}_{rank}^P \rfloor = \lfloor \bar{\bar{j}}_{rank}^P \rfloor \text{ and } (\bar{j}_{sub-rank}^P < \bar{\bar{j}}_{sub-rank}^P) \right) \end{aligned} \quad (4.12)$$

In this way we cluster the points according to the probabilistic crowding distance (i.e. preserving diversity) while ensuring the use of the additional informations obtained from uncertainty quantification (i.e. tight coupling).

Deterministic consistency: as we mentioned earlier, when all the probability distributions are deterministic the probabilistic partial order turns in the deterministic partial order

$$\prec_n^P \xrightarrow{\frac{\partial f(z,\xi)}{\partial \xi} = 0, \forall \xi \in \Omega} \prec_n \quad (4.13)$$

where

$$\begin{aligned} \bar{j} \prec_n \bar{j} \quad & \text{if } (\bar{j}_{rank} < \bar{j}_{rank}) \\ & \text{or } ((\bar{j}_{rank} = \bar{j}_{rank}) \text{ and } (\bar{j}_{distance} < \bar{j}_{distance})) \end{aligned} \quad (4.14)$$

considering the original deterministic definitions of the rank, j_{rank} , and the crowding distance, $j_{distance}$.

4.3 The algorithm

The presented probabilistic non dominated sorted genetic algorithm (**P-NSGA**) is organized in several steps that consist of:

1. **Initialization of the population.** The initial population is provided with a random seeding in the subset of the design space, Z ;
2. **Sampling and objective evaluations in the initial population.** For each candidate design an ensemble of calculations is obtained by sampling the probability space, Ω , using an uncertainty quantification methodology (e.g. Monte Carlo). The evaluations of the objective functions or the corresponding estimations using the different methodologies (e.g. Simplex Stochastic Collocation) are stored for each candidate design. It is interesting to notice that these samples are generally used to approximate the statistical moments (i.e. compute integrals) in all the approaches described in Section 4.1 and then erased, while in the present methodology they are stored by the algorithm and used in the sorting procedure;

3. **Probabilistic non-dominated sorting (initial population).** The probabilistic rank and the probabilistic sub-rank are used to sort the parent population using the objective evaluations at the samples;

Start of the evolution process (the following actions are performed in each generation)
4. **Selection of the parents.** Parents are selected for reproduction to generate offspring. In the same fashion of the original NSGA-II we use a binary tournament selection based on the probabilistic crowded-comparison operator. Tournament selection is carried out until the pool (i.e. the number of parents to be selected) size is filled,
5. **Crossover and mutation.** The selected population generates offsprings from crossover and mutation operators (e.g. the original NSGA-II algorithm uses Simulated Binary Crossover (SBX) and Polynomial mutation);
6. **Sampling and objective evaluations (offspring population).** As in step 2 but using the offspring population;
7. **Probabilistic non-dominated sorting (intermediate population).** The offspring population is combined with the current generation population. The probabilistic rank and the probabilistic sub-rank are used to sort this intermediate population using the objective evaluations at the samples;
8. **Selection of the novel population.** The new generation is filled by

each front subsequently until the population size exceeds the current population size using the probabilistic crowded-comparison operator. If an individual does not survive from a generation to the next one its evaluations of the objective functions at the samples are overwritten and the corresponding memory reallocated;

End of the evolution process

9. Post-processing of the Pareto Front

4.4 Multi-objective analytic test function

In this section the effectiveness of the effectiveness of the probabilistic Crowded-Comparison Operator introduced in Section 4.2.3 is verified on an analytic multi-objective test case, namely the Fonseca-Fleming function [84]. The Fonseca's two-objectives minimization problem is characterized by a non-convex Pareto optimal front and a large and non-linear trade-off curve. The two-objective functions, $f_1(\vec{x})$ and $f_2(\vec{x})$, to be minimized are given as

$$\begin{aligned} f_1(\vec{x}) &= 1 - \exp\left(-\sum_{i=1}^N \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \\ f_2(\vec{x}) &= 1 - \exp\left(-\sum_{i=1}^N \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \end{aligned} \quad (4.15)$$

where taking x_1 and x_2 to be design variables with bounds $[-4, 4]$ and taking x_3 to be a standard normal random variable $\mathcal{N}(\mu = 0; \sigma = 0.2)$ it is possible to solve the multi-objective problem in presence of uncertainty.

The proposed P-NSGA has been used to obtain the probabilistically non-dominated shown in Figure 4.2 while the original NSGA-II algorithm has been used to obtain the deterministic Pareto front, used as reference solution.

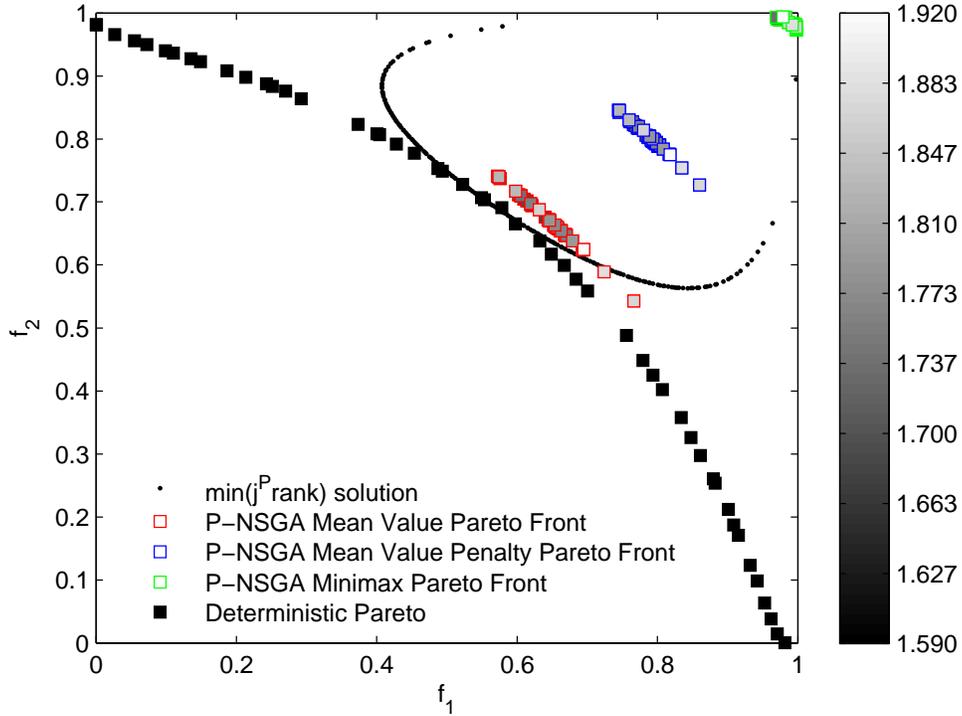


Figure 4.2: Non dominated solutions obtained with P-NSGA in comparison with deterministic Pareto.

It is interesting to notice that since all the informations about the non-dominated solutions are stored by the algorithm all the Pareto fronts (e.g. Mean Value Pareto, Mean Value Penalty Pareto, etc.) can be generated a-posteriori, while the other methods in Section 4.1 use only partial integral informations (e.g. several statistical moments, etc.).

The non-dominated solutions have been colored by the probabilistic rank, which could be either used as an additional selection criterion on the various Pareto fronts.

The ensemble of solutions of the individual characterized by the lowest probabilistic rank (e.g. the closest to the Pareto dominance for each value of the uncertain variable) has been scattered in the same figure

In Figure 4.2 it is possible to show that the Pareto fronts generated a-posteriori with P-NSGA fall on the corresponding fronts obtained using the NSGA-II algorithm with the mean (Mean Value Pareto) or the mean plus the standard deviation (Mean Value Penalty) as objectives. The same random state (e.g. generation of random numbers) has been supposed in the sampling procedure to give sense to the comparison among the different Pareto fronts, hence the difference among the fronts is only due to the selection process or objective evaluation and doesn't rely on randomness.

In Figure 4.4 we reported the cumulative distribution functions with error bounds calculated using Greenwood's formula. Further details about the optimization strategy can be found in Table 4.1.

Table 4.1: Multi-objective optimization strategy

Parameter	Value
Population size [-]	50
Crossover fraction [-]	0.90
Mutation fraction [-]	0.10
Parent sorting	Tournament between couples
Mating Pool [%]	50
Crossover mode	Simulated Binary Crossover (SBX)
Generations [-]	300
Uncertainty Quantification	LHS
Number of samples [-]	200

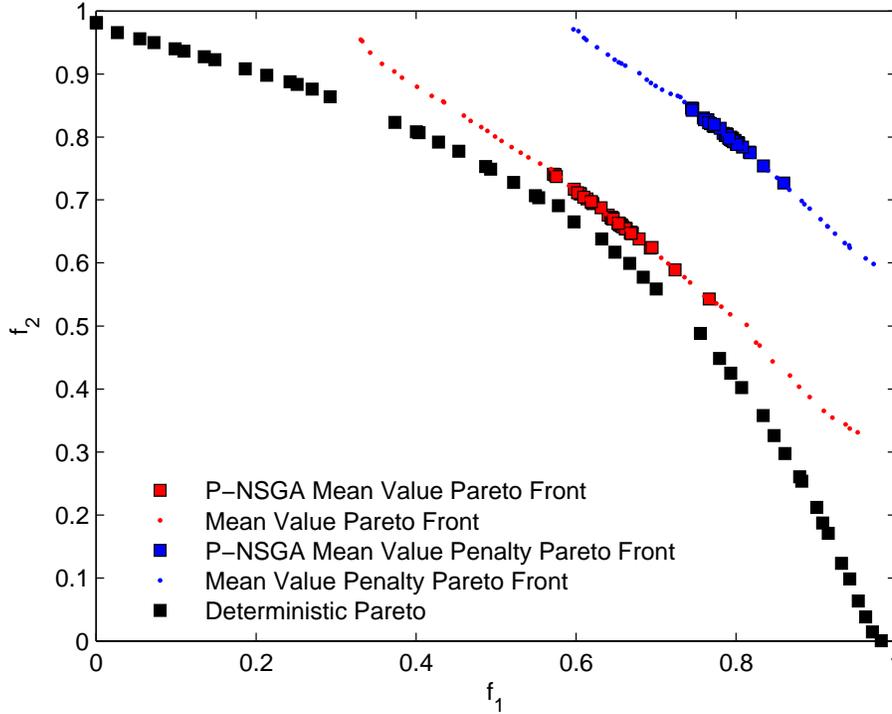


Figure 4.3: Non dominated solutions obtained with P-NSGA in comparison with Mean Value Pareto and Mean Value Penalty Pareto.

4.5 Case study: Multi-Objective Optimization under Uncertainty of a Wind Turbine

In this section we consider the AOC 15/50 wind turbine that was analyzed in presence of uncertainties in chapter 1. The insect contamination is considered as unique source of uncertainty, while the shape of the cross-sections and the twist and chord distributions are optimized under uncertainty using the framework presented in the previous sections.

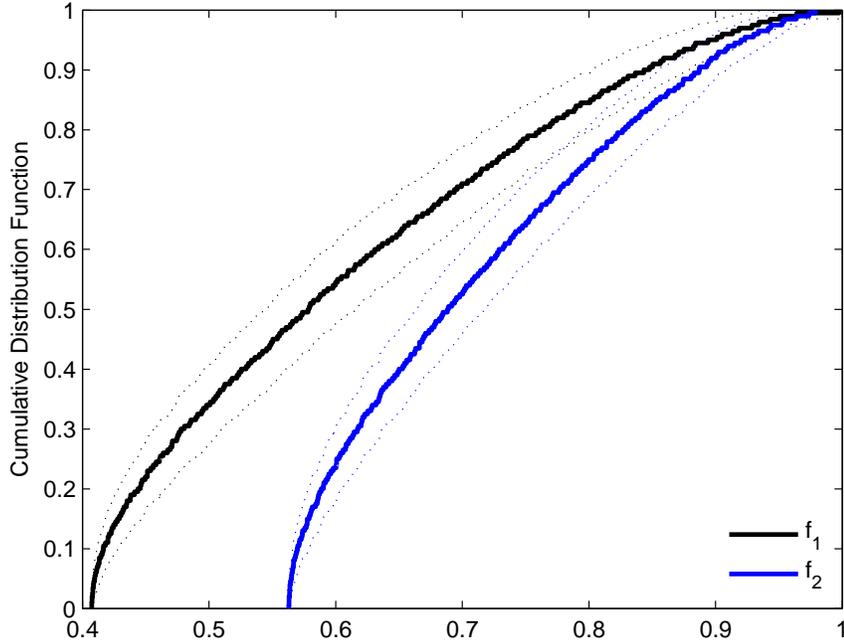


Figure 4.4: CDFs of the objectives for the solution closest to probabilistic Pareto dominance.

4.5.1 Definition of the shape optimization problem

We follow Zhong and Qiao’s work [85] and use B-splines to parameterize the geometry. Specifically, we consider fifth order B-splines with a nominal uniform knot set to represent both the cross-sections of the turbine blades and the distribution of the chord and twist along the span. The geometry of the base airfoils is given by the following equations

$$f = f_0 + \sum_1^{n_f} P_i N_i(X) \quad (4.16)$$

$$g = g_0 + \sum_{n_f+1}^{n_f+n_g} P_i N_i(X) \quad (4.17)$$

where f and g are the upper and bottom surface, f_0 and g_0 are the initial bottom and upper surface, P_i are the control points of the B-splines and $N_i(X)$ are the B-spline basis functions. Three geometrical constraints are enforced: the first is to avoid intersections of the upper and lower airfoil surfaces, the second is to reduce the changes in curvature in either the upper or lower surface of the airfoil, and the third is for to enforce a maximum thickness of the airfoils.

The chord and twist distributions are also parametrized similarly using B-splines:

$$\theta = \theta_0 + \sum_1^{n_\theta} P_i N_i(X) \quad (4.18)$$

$$chord = chord_0 + \sum_1^{n_{chord}} P_i N_i(X) \quad (4.19)$$

where θ and $chord$ are the distribution of twist and chords after optimization, θ_0 and $chord_0$ are the initial distribution of twist and chords, P_i are the control points of the B-splines and $N_i(X)$ are the B-spline basis functions. The degree of the B-splines chosen for twist and chord optimization is three. It should be noted that when all the control points of the airfoil parametrization and of the chord and twist distribution have zero values the original shape is returned.

The optimization process (in the absence of uncertainties) uses the NSGA-II algorithm that considers as objectives the maximization of the averaged power coefficient over the last minute of the simulation [-] and the minimiza-

tion of the Overall Sound Pressure Level [dB] at the observer location.

4.5.2 Deterministic optimization

An initial multi-objective optimization (maximize the power coefficient while minimizing the noise) has been carried out ignoring the insect contamination and the resulting Pareto front is shown in Figure 4.5. The baseline blade was already optimized by the manufacturer but due to the steep characteristics of the Pareto at high power coefficient it was possible to find a trade-off design with considerable less noise. The population of the GA contained 40 individuals, which evolved for 50 generations.

Successive simulations were performed accounting for uncertainties due to insect contamination.

4.5.3 Optimization under Uncertainty

The deterministic Pareto front was used as initialization for the procedure proposed in the previous section. In Figure 4.6 a close-up of the design space close to the previous trade-off design is shown. It is important to notice that in the presence of uncertainty each new design is actually stochastic and characterized by a number (cloud) of probable results. We refer to the Best Efficient Point (BEP)[80] (i.e. the knee of the Pareto Front) in the presence of uncertainty as the ROpt (Robust Optimum) design.

It is interesting to note that the ROpt design does not lie on the deterministic Pareto front and conversely that the deterministic trade-off design does not lie on the probabilistic Pareto front. The RI for the ROpt design

assumes the lowest value in the population and has been chosen as a decision making criterion for the probabilistic Pareto front, see Figure 4.6. Moreover, the deterministic trade-off design has been chosen as the BEP of the deterministic Pareto front.

The spread of the ROpt designs suggests a stable solution in the presence of uncertainty since it is more probable that a design is located at higher values of the averaged power coefficient and the distribution is rather uniform compared to the trade-off spread, as shown in Figure 4.7.a. Additionally, the spread of Overall Sound Pressure Level of the ROpt design is clearly dominating the deterministic trade-off, as shown in Figure 4.7.b. It is interesting to notice that the presented novel method returns as result of the process of robust optimization the optimal design variables and the CDFs of the objective functions: the CDF represents a complete information in probability about the behavior of the candidate design rather than few real value attributes given by the other methods discussed in Section 4.1.

The optimized design (red) reveals a significant increase in the length of the chords at the inner part of the blade, while its twist follows a smoother distribution than the baseline (black) showing higher torsion closer to the root and tip sections, as shown in Figure 4.8.b. The optimized airfoils for the ROpt design are shown in Figure 4.9 in order to compare them with the baseline solution. It is interesting to notice the change in the trailing edge at the root section of the blade, which could be related to the noise reduction and the significant change in curvature of the mid-span airfoil. The tip airfoil reveals a significant increase in thickness up to 60% of the

chord, with a trailing edge very similar to the baseline solution.

4.6 Lessons learned

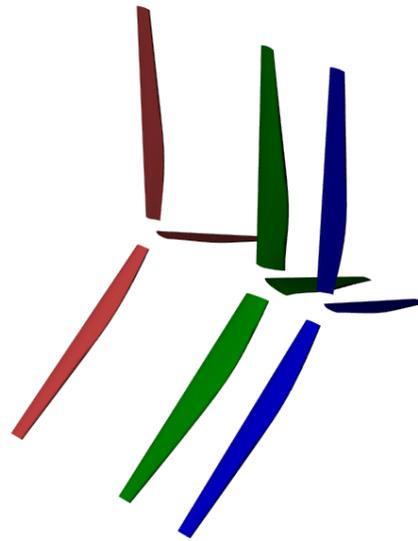
In this chapter it was proposed a novel intrusive approach to couple multi-objective optimization and uncertainty quantification in order to let the latter be a driver in the shifting of the Pareto Front due to uncertainties. The proposed approach is based on the full probabilistic description of the objective function under uncertainty and appears promising at overcoming the main drawbacks of other methods found in literature, even if further benchmarks are required. The deterministic NSGA-II algorithm was generalized and a probabilistic counterpart, the P-NSGA, was obtained. It was shown that, since all the informations about the non-dominated solutions are stored by this algorithm, all the Pareto fronts (e.g. Mean Value Pareto, Mean Value Penalty Pareto, etc.) can be generated a-posteriori.

In the end of the chapter a novel process of optimization of wind turbine blades under uncertainty was presented. The design obtained with the novel procedure appears to be more stable in the presence of uncertainty than its deterministic counterpart, hence the main goal of this case study has been successfully achieved.

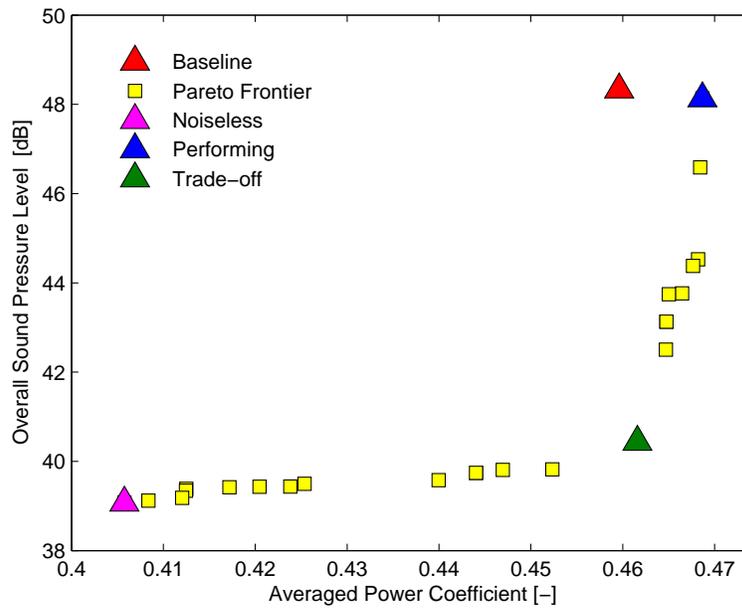
All the objective evaluations concerning the wind turbines and the F1 wheel assembly (see chapter 2) were carried out on an high performance cluster (Certainty, Stanford University). When we're in presence of computer intensive extreme ensemble computations for uncertainty quantification and optimization under uncertainty, we need to perform them in the most efficient

CHAPTER 4. MULTI-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY

way. Indeed the computational resources needed to consider the uncertainty quantification in an optimization procedure require at least an increase in an order of magnitude in the number of evaluations of the response of the system with respect to deterministic optimization. This motivate the effort spent on High Performance Computing in the next chapter of this thesis.

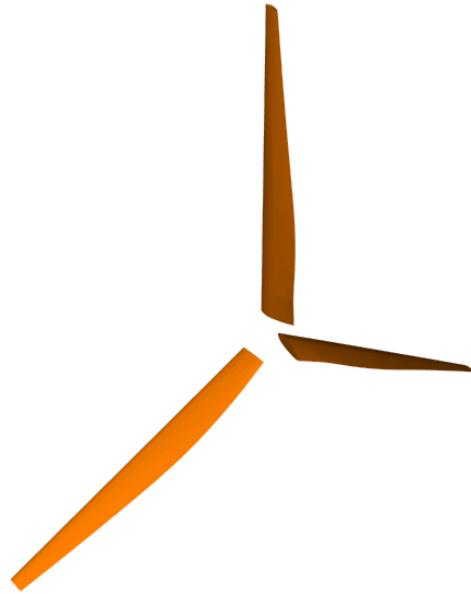


(a) Baseline, Trade-off and highest Performing designs

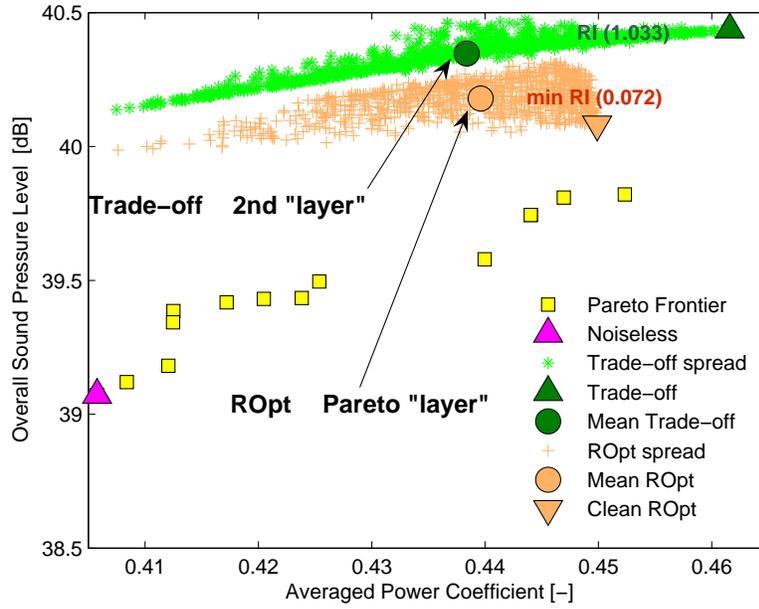


(b) Non-dominated solutions

Figure 4.5: Deterministic Pareto front

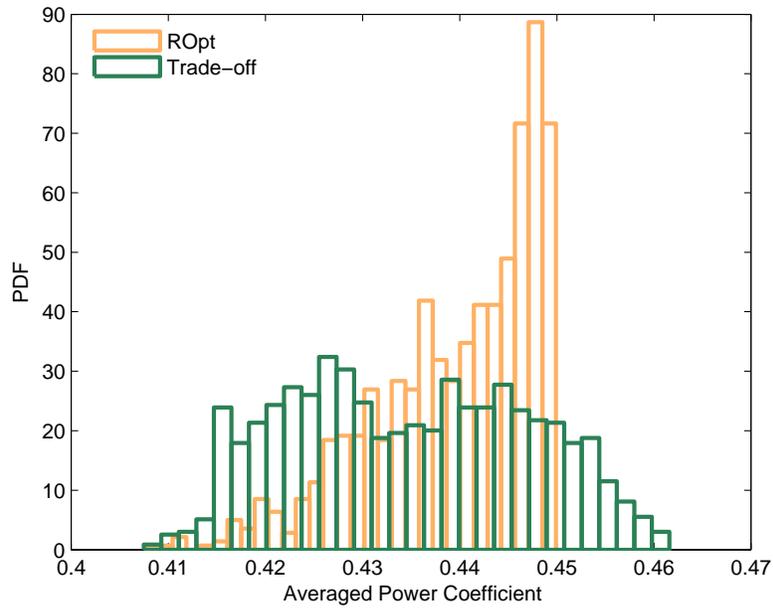


(a) ROpt design

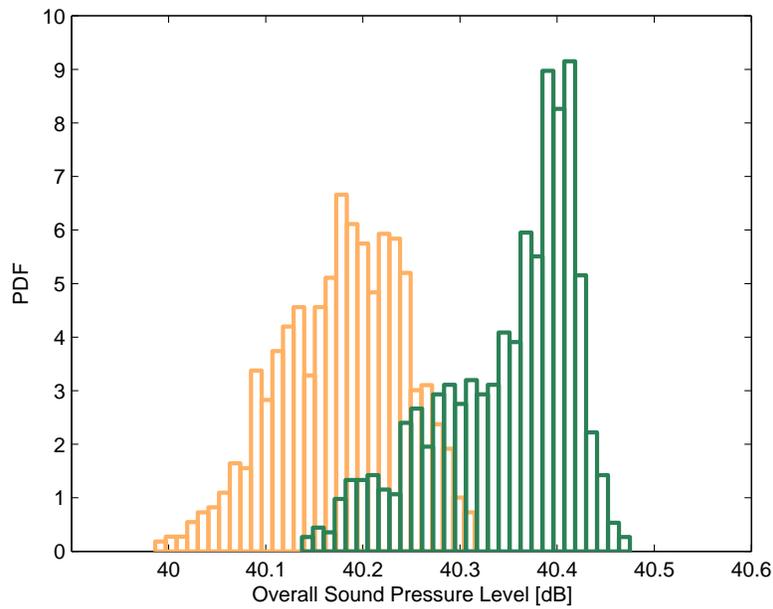


(b) Probabilistic Pareto front (detail)

Figure 4.6: Probabilistic Pareto front



(a) PDF of Averaged Power Coefficient



(b) PDF of Overall Sound Pressure Level

Figure 4.7: Probabilistic and deterministic trade-off designs

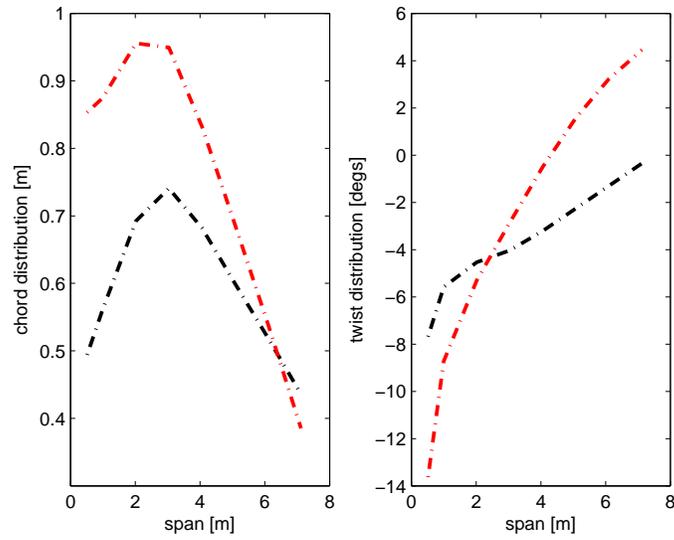


Figure 4.8: Chord and twist distributions

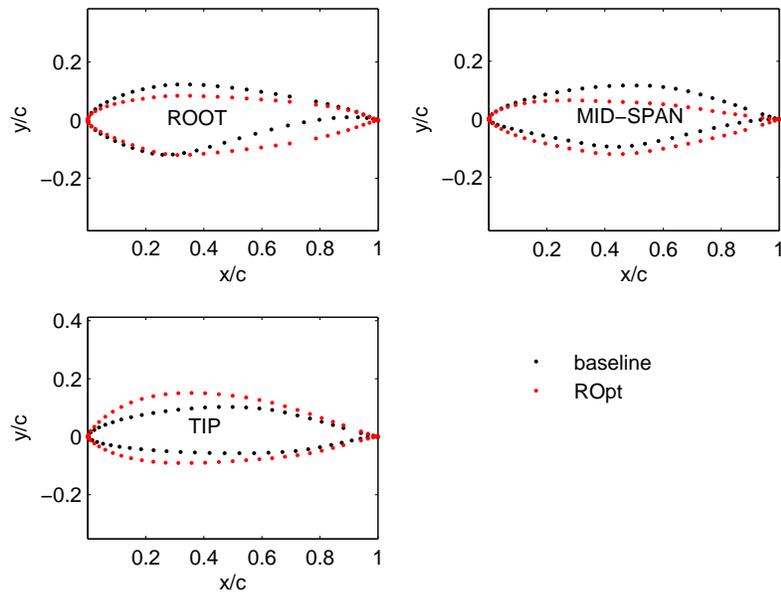


Figure 4.9: Wind turbine cross-sections

Chapter 5

Extreme Ensemble Computations

In the previous chapters it was shown that the development of robust design strategies coupled with detailed simulation models require the introduction of advanced algorithms and computing resource management tools. A simulations environment – Leland – has been developed to dynamically *schedule, monitor and stir* the calculation ensemble and extract runtime information as well as simulation results and statistics. Leland is equipped with an auto-tuning strategy for optimal load balancing and fault tolerance checks to avoid failures in the ensemble.

5.1 The need of High Performance Computing

Efficient algorithms to tackle optimization and uncertainty of computationally intensive simulations are becoming increasingly important due to the availability of high-performance computing clusters. In the last few years, clusters with 10,000 CPUs have become available, and it is now feasible to design complex engineering systems in such an environment. This develop-

ment highlights the need to create robust algorithms and to design resource managers that deliver cost-effective utilization with fault tolerance. The BlueGene/L cluster with 65,536 nodes was designed to have less than one failure every ten days. In fact, this cluster and others like it experience an average of one processor failure every hour [86]. In light of this, it is necessary to study, develop, and continually improve strategies for efficient completion of large applications. Theoretical work has been published in the literature that suggests that advanced algorithms need to be implemented [87]. But a majority of this work deals with test functions on a small number of compute nodes.

The design process could involve running an extreme number of large computations or 'extreme ensemble' (on the order of thousands) in order to create a robust solution that will remain optimal under conditions that cannot be controlled ('uncertainties'). The ensemble is a list of runs generated by the optimization and uncertainty algorithms that is dynamic in nature and is not deterministic. This means that the number of additional simulations is dependent on the results of the prior converged simulations. For the study shown in chapter 2, there are approximately 400 simulations to perform per optimization cycle (i.e. generation). When the results of those 400 simulations are analyzed, an additional list of 400 simulations, each with a unique range of input parameters, are generated for the next generation in the optimization process. The values of the the input parameters for the next generation cannot be known a priori.

5.2 Leland

It was developed a simulations environment, hereafter referred to as Leland, that allows us to monitor the calculation ensemble and extract runtime information as well as simulations results and statistics on the fly. Leland is equipped with an auto-tuning strategy for optimal processor count selection and fault tolerance to ensure that a simulation or a processor stall is detected and does not impact the overall ensemble finish time. The structure of Leland is based on a workflow through I/O sub-systems that represent the software applications (i.e. Sculptor, Fluent, Tecplot, Matlab etc.) involved in the process. This environment is designed to run natively on any high-performance computing (HPC) system, by integrating with the job-submission/queuing system (e.g. Torque). Moreover, it does not require administrator permissions: once the analysis is initiated multiple simulations are submitted and monitored automatically. In Leland, a job is an instance of the entire multi-physics simulations, which might include grid generation, mesh morphing, flow solution and post-processing.

The main objective of Leland is to set-up a candidate design as a job and to manage it until it is completed and to gather relevant results that are used to inform the robust optimization process. ROpt (robust optimization), shown in figure 5.1a, is the engine behind this design environment. Given the design and/or uncertain input variables, ROpt continuously generates new design proposals (samples) based on the evolutionary strategy and/or analysis of the uncertainty space, until a convergence criterion is met. The Job Liaison, shown in Figure 5.1b, defines the characteristics of each single job

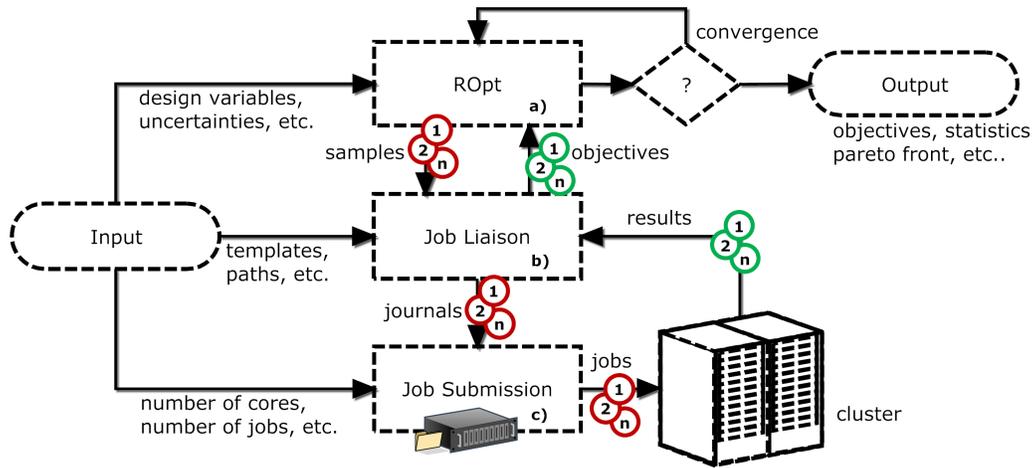


Figure 5.1: Leland flowchart

and continuously monitors the progress of the simulations until completion in order to communicate the objective evaluations back to ROpt. It is the job of this module to continuously monitor for faults, stalls, or errors to ensure that the total runtime is not detrimentally affected by processor/memory failure. The Job Submission engine, shown in Figure 5.1c, ensures that the correct number of jobs is always running on the cluster. The variables (number of cores, number of jobs, etc.) from the input file that are used to initialize the runs are dynamic, meaning they can be edited on the fly and the system will respond accordingly.

Leland has the ability to dynamically select the optimal number of processors to run per realization. This is achieved by auto-tuning. The user selects an optimal window of cores to use per realization prior to launching the full ensemble. The auto-tuning algorithm then samples the space by using a unique number of cores per realization in the ensemble. Once

two or more realizations are complete the auto-tuning algorithm can start to construct an application specific speed-up curve (Figure 5.2). Speed-up is defined as the total time required to finish the simulation using 1 processor divided by the total time required to finish the simulation using p processors

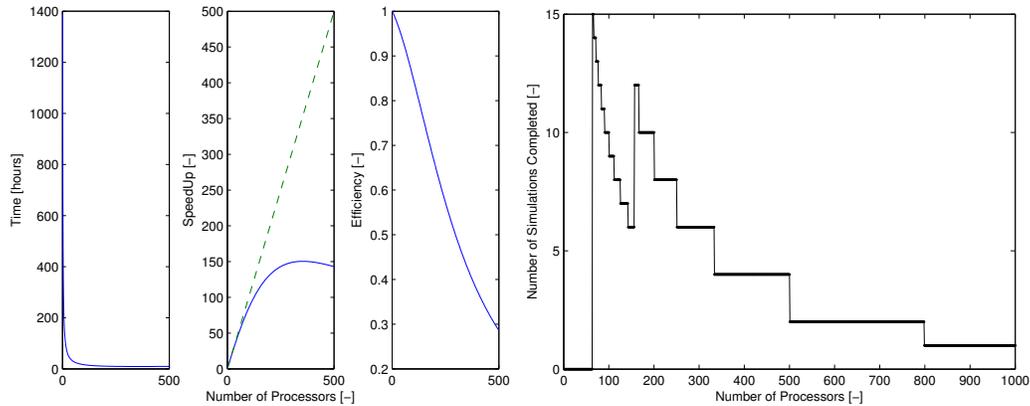
$$\text{Speedup}(p) = \frac{t_{total}(1)}{t_{total}(p)} \quad (5.1)$$

The speed-up curve in Figure 5.2 was generated by artificially replicating an HPC simulation. The time required to complete an HPC simulation is primarily a function of three factors i) portion of the code that is not parallelizable (t_{serial} in Equation 5.2) ii) portion of the code that is parallelizable ($t_{parallel}$ in Equation 5.2) and iii) the communication time between CPUs (t_{comm} in Equation 5.2)

$$t_{total} = t_{serial} + t_{parallel} + t_{comm} = \quad (5.2a)$$

$$= 5000 + \frac{5 \times 10^6}{p} + 40p \quad (5.2b)$$

The serial portion of code in the example shown in Figure 5.2 is constant (5000 seconds) and not a function of the number of processors allocated to the job. The length of time required to complete the parallel portion of code in the example shown in the same figure is 5 million seconds divided by the number of processors used. Finally, there will always be some latency between CPUs and this is characterized by the communication time between nodes. The linear penalization we used in this example is 40 seconds per processor, but the latency slowdown could also be a more complex function related to the specific application.



(a) Total time required to complete simulation as a function of the number of processors (left), Speed-up curve (middle), and Efficiency curve (right) (b) Number of simulations that would be completed in a 24 hour window with 1000 available processors using exactly p processors for each simulation

Figure 5.2: Sample HPC simulation diagnostics

Linear speed-up, also referred to as ideal speed-up, is shown as the green dotted line in the middle plot of Figure 5.2. An algorithm has linear speed-up if the time required to finish the simulation halves when the number of processors is doubled. It is common for fluid dynamic simulations to experience speed-down; this occurs when the total time required to finish the simulation actually increases with increasing processors. Leland has the ability to recognize the point at which speed-down occurs (at about 400 processors in Figure 5.2) and never use more than this number of processors. The rightmost plot in Figure 5.2 shows the efficiency (defined by Equation 5.3) curve for this artificial HPC simulation. The efficiency typically ranges between values of $0 \sim 1$ estimating how well utilized the processors are compared to the effort

wasted in synchronization and communication

$$\text{Efficiency}(p) = \frac{\text{Speedup}(p)}{p} \quad (5.3)$$

It is clear from the previous plot that the highest efficiency occurs with the lowest number of processors. This speed-up curve will guide Leland's auto-tuning algorithm in assigning the optimal number of cores per realization (which may not be in the users original window). Since an ensemble of this size takes more than a few weeks on a large cluster, multiple job submissions need to be submitted to the local queuing system. These jobs are typically limited to 24 hour run times (or a wall clock time of 24 hours). Thus, it is essential that the auto-tuning algorithm recognizes how many hours remain prior to the job terminating due to the wall clock time and tries to increase the number of cores to finish as many realizations as possible within a specific time frame.

5.3 The input file

Leland is totally driven by a unique input file in which is possible to define all the variables needed by the queue manager, the job liaison and the optimization and/or uncertainty quantification algorithm. In Figure 5.3 it is reported the input file used to perform uncertainty quantification using EOLO as executable (see chapter 1).

The input file requires different informations to perform a calculation using the Leland platform. There is the need to specify an user name that will be used to create a process on the cluster (e.g. gpetrone).

CHAPTER 5. EXTREME ENSEMBLE COMPUTATIONS

```
# Dynamic Input File for LELAND - EOLO used as executable

# Variables for leland
user          gpetrone          # Username on Cluster
cluster_jobs  1                 # Number of cluster jobs
nodes_per_job 1                 # Number of nodes allocated on the cluster for each job
cores_per_node 24              # Number of cores on each node (depends on machine architecture)
cores_used_per_node 24         # Number of cores to use for each node
cores_per_run 1                 # Number of cores for each individual simulation
job_name      SSS_complete      # The name of the cluster jobs
run_name      CORE              # The identifier for each simulation (i.e. run_name_1 will be the first simulation,etc.)
executable    job_files/eolo.sh # The script that contains the actions to perform a simulation
matlab_directory matlab_files  # The directory where the Matlab files are stored
job_directory job_files         # The directory where leland infrastructure exists
output_directory output_files  # The directory where the output files are stored
journal_template journal_files/runEOLO_input.dat # The path and name of journal template
wall_clock_time 23:59:59       # Wall clock time for each job
queue_type     default          # The type of queue requested (Most typically either default or debug)

# Variables for ROpt
objective      power_coefficient # Objective function
objective      OSPL              # Objective function
optimization   NONE              # Optimization flag (METHOD/NONE)
UQ             SSC              # Uncertainty Quantification flag (METHOD/NONE)
n_sample_max   90                # Max number of simplex refinements
epsilon_stop   1e-3              # Simplex convergence threshold
refinements_per_time 15          # Number of simultaneous simplexes refinements
uncertain_variable uniform      n_critical_root 1 9 # Uncertain variable type, name and attributes
uncertain_variable uniform      n_critical_mid  1 9 # Uncertain variable type, name and attributes
uncertain_variable uniform      n_critical_tip  1 9 # Uncertain variable type, name and attributes
```

Figure 5.3: Sample input file for Uncertainty Quantification

Several inputs of Leland are dynamic in nature, this means that they can be changed on the fly and the whole system will respond accordingly, as shown in Figure 5.4.

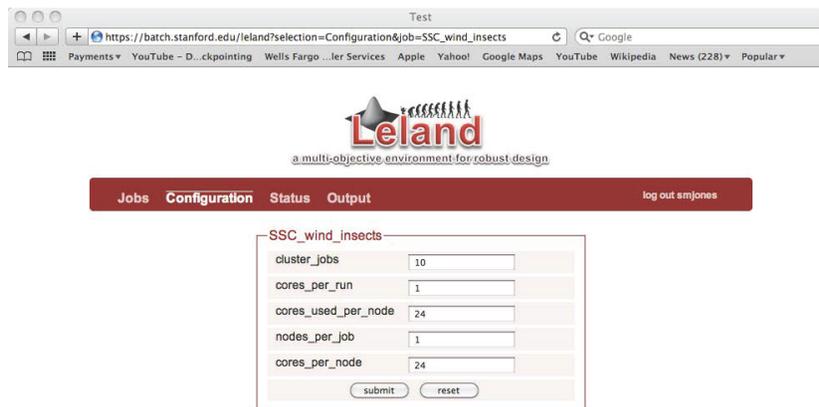


Figure 5.4: Web interface for dynamic inputs

The name of the job is used to create a job on the cluster, while the core name will be the identifier associated to that job. It is important to point to the executable – a script containing all the actions needed to perform a simulation. For example a script can be made of a process of grid generation, mesh morphing and CFD analysis; in this case two or more softwares are invoked.

The logic behind Leland is based on having a journal template, where the variables are bounded by exclamation points (e.g. !variable!) and which will be replaced with the samples given by the optimization or uncertainty quantification algorithm. This process allows the creation of an ensemble of runs on the cluster, as shown in Figure 5.5.

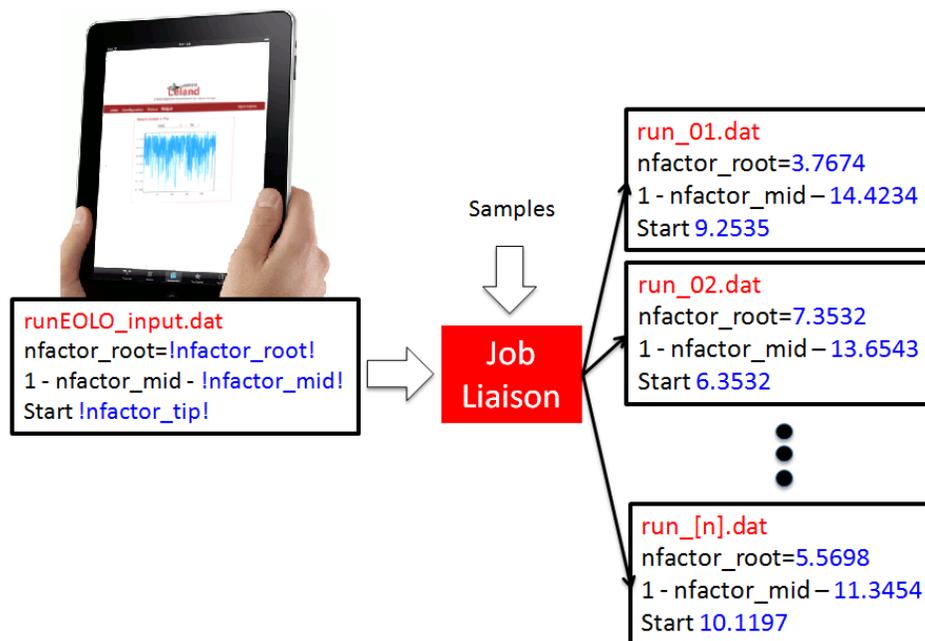


Figure 5.5: The job liaison

CHAPTER 5. EXTREME ENSEMBLE COMPUTATIONS

The second part of the input file is totally dedicated to the algorithms used to generate the samples. Leland searches the output directory looking for the objectives specified (e.g. power coefficient, noise, etc.). The algorithms actually codified are able to perform uncertainty quantification, deterministic optimization or optimization under uncertainty by using the methods given by the entries *UQ* and *optimization*. In this example the SSC algorithm is used to sample up to 90 samples (using a 15 by 15 parallel refinement) until an accuracy of $1e-3$ is obtained in the error estimate. The uncertain variables are specified as uniform distribution ranging from 1 to 9 (insect contamination).

The web interface for the power coefficient is reported in Figure 5.6, where a negative sign is used according to an optimization convention.

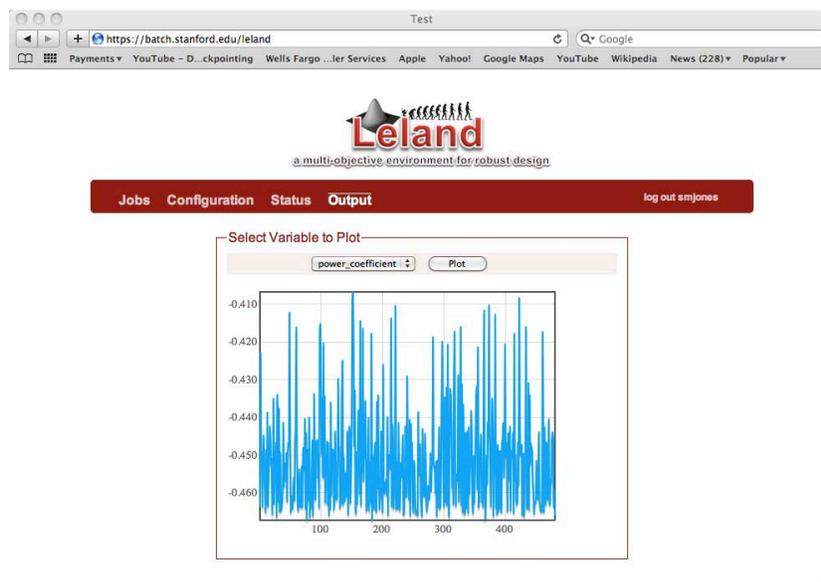


Figure 5.6: Web interface for objective functions

CHAPTER 5. EXTREME ENSEMBLE COMPUTATIONS

In Figure 5.7 we reported the input file used to perform Optimization Under Uncertainty using the commercial softwares Sculptor (Optimal Solutions) and Fluent (Ansys) as executables. In this case 24 processors are reserved to each simulation, while in the case of EOLO each simulation was carried out on a single processor. The optimization algorithm (NSGA-II with 22 individuals evolving for 15 generations) uses deterministic attributes of the objective functions as objectives. Indeed the entries labeled as *coupling* define the way in which the optimization algorithm is coupled with the uncertainty quantification algorithm (LHS with 11 samples). In this case the mean plus the standard deviation (e.g. *PENALTY+*) has been used as deterministic attribute of any objective. The label *P-NSGA* would have activated the coupling between optimization and uncertainty quantification introduced in chapter 4.

```
# Variables for leland
user          axerio          # Username on Cluster
cluster_jobs  5                # Number of cluster jobs
nodes_per_job 20              # Number of nodes allocated on the cluster for each job
cores_per_node 24             # Number of cores on each node (depends on machine architecture)
cores_used_per_node 12        # Number of cores to use for each node
cores_per_run 24              # Number of cores for each individual simulation
job_name      leland          # The name of the cluster jobs
run_name      f1              # The identifier for each simulation (i.e. run_name_1 will be the first simulation, etc.)
executable    job_files/fl.sh # The script that contains the actions to perform a simulation
matlab_directory matlab_files # The directory where the Matlab files are stored
job_directory job_files      # The directory where leland infrastructure exists
output_directory output_files # The directory where the output files are stored
journal_template parallel_files/parallel_template.jou # The path and name of journal template
journal_template serial_files/serial_template.jou   # The path and name of journal template
journal_template sculptor_files/sculptor_template.sh # The path and name of journal template
journal_template iterating_files/iterating_template.sh # The path and name of journal template
wall_clock_time 23:59:59      # Wall clock time for each job
queue_type      default       # The type of queue requested (Most typically either default or debug)

# Variables for Ropt
objective       drag_force    # Objective function
objective       mass_flow     # Objective function
optimization    NSGA2        # Optimization flag (METHOD/NONE)
UQ              LHS          # Uncertainty Quantification flag (METHOD/NONE)
gen             15            # Number of generation for optimization
pop             22            # Size of population for optimization
n_sample_max    11           # Max number of simplex refinements [used only if Uncertainty Quantification flag is active]
design_variable  bd_width     -0.01  0.01  # Design variable name and range
design_variable  bd_height    -0.01  0.01  # Design variable name and range
design_variable  bd_length    -0.01  0.01  # Design variable name and range
uncertain_variable uniform   cp_width -0.02  0.015 # Uncertain variable type, name and attributes
coupling_1      PENALTY+     # Coupling between Optimization and Uncertainty
coupling_2      PENALTY+     # Coupling between Optimization and Uncertainty
```

Figure 5.7: Sample input file for Optimization under Uncertainty

5.4 The future of Leland

Leland is actually used to perform extreme ensemble computations at the Mechanical Engineering and Institute for Computational Mathematical Engineering and the Center for Turbulence Research (CTR) located at Stanford University (CA, United States). There are several users which use Leland with different focuses, spacing from chemistry to hypersonic simulations and from wind turbines design to intensive CFD simulations of Formula 1 subsystems. In the next years Leland will be adopted as meta-scheduler in several national laboratories in US and on the platform S.C.O.P.E. of the University Federico II of Naples.

Chapter 6

A bridge with games

The language of Game Theory – coalitions, payoffs, markets, votes – suggests that it is not a branch of abstract mathematics; that it is motivated by and related to the world around us; and that it should be able to tell us something about that world.

— R.Aumann

In the previous chapter it was introduced a methodology to optimize in presence of uncertainties. In this chapter Game Theory is used to define a process of optimization under uncertainty as a conflict between players. This novel approach could be particularly useful in presence of incomplete informations about the aleatory uncertainties or as a particular way to handle epistemic uncertainties. Indeed the classic concepts of Nash and Stackelberg equilibria are formulated in terms of entropy, a concept that comes from information theory and measures the content of informations of a probabilistic distribution. The chapter starts with a brief review of the concepts of Game

Theory, Section 6.1, then the novel procedure is presented in Section 6.2. An extension of the present model to multiple leaders and/or multiple followers is presented in Section 6.3 while the algorithms are presented in Section 6.4. A further extension to challenge the non-uniqueness of the solution of the game is presented in Section 6.5.

6.1 Game Theory

Game Theory is the formal study of conflicts and cooperation. Game theoretic concepts apply whenever the actions of several agents are interdependent. These agents maybe individuals, groups, firms, or any combination of these. The concepts of game theory provide a language to formulate, structure, analyze, and understand strategic scenarios. As a mathematical tool for the decision-maker the strength of Game Theory is the methodology it provides for structuring and analyzing problems of strategic choice. The process of formally modeling a situation as a game requires the decision-maker to enumerate explicitly the players and their strategic options, and to consider their preferences and reactions.

6.1.1 Definition of game

A game is a situation that involves two or more decision makers, the players, where each player tries to minimize or maximize his objective (cost or profit) and each objective depends not only on the option that he chooses but also on the options chosen by the other players. We speak about strategic interaction. Let $\Gamma = \langle n; X_1, X_2, \dots, X_n; f_1, f_2, \dots, f_n \rangle$ be an n-person normal

form game with player set $I = 1, 2, \dots, n$, with strategy space X_i for each player $i \in I$ and where $f_i : X^1 \times \dots \times X^n \rightarrow R^n$ is the payoff function of player $i \in I$. If player i chooses $x_i \in X$, then he obtains a cost $f_i(x_1, \dots, x_n)$.

6.1.2 The Nash equilibrium

A desirable game-theoretic solution is one in which individual players act in accordance with their incentives, minimizing or maximizing their own payoff [88, 89, 90]. This idea is captured by the notion of Nash equilibrium strategy [91]: each player optimizes his payoff and no single player can individually improve his welfare by deviating. So, the Nash equilibrium strategy acquires the notion of a stable solution. We denote by x_{-i} the vector $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. If non-cooperative behavior is assumed between the n players, the equilibrium solution considered is the well known concept of Nash equilibrium. A Nash equilibrium is a vector (x_1^N, \dots, x_n^N) such that for each $i \in I$

$$f(x_1^N, \dots, x_n^N) = \inf_{y \in X_i} f(x_1^N, \dots, x_{i-1}^N, y, x_{i+1}^N, \dots, x_n^N) \quad (6.1)$$

Unfortunately, a Nash equilibrium may not exist, may not be unique and also may not be optimal for the players. It turns out to be very hard sometimes to check if the game has a Nash equilibrium and in this case to compute it as suggested in the Nash's theorem proof by using a fixed point theorem.

6.1.3 The Stackelberg equilibrium

We deal with a two-person normal form game $\Gamma = \langle 2; X, Y; l, f \rangle$. One of the players called leader has the leadership in playing the game because he

is informed on the opponent's best reply to any possible his decision. This opponent is called follower. X, Y are subsets of metric spaces, and represents respectively the leader's and the follower's strategy set. l, f are real valued functions defined on $X \times Y$ and represent respectively the leader's and the follower's cost functions. Both players are cost minimizing. The concept of Stackelberg equilibrium goes back to von Stackelberg [92] who introduced it in the context of duopolistic competitions. Here we recall the definition.

Let $x \in X$ be the leader's announced choice. For any $x \in X$, the follower solves the following lower level problem, $\mathcal{P}(x)$

$$\begin{cases} \text{find } \bar{y} \in Y \text{ such that} \\ f(x, \bar{y}) = \inf_{y \in Y} f(x, y) \end{cases} \quad (6.2)$$

The solution to this problem is supposed to be unique and denoted by $\tilde{y}(x)$. The function $x \in X \rightarrow \tilde{y}(x) \in Y$ is called follower's best reply or best response. The leader knows this best reply and optimizes his own updated cost function $l(x, \tilde{y}(x))$, just considering the follower's response and solves the following upper level problem, \mathcal{S}

$$\begin{cases} \text{find } \bar{x} \in X \text{ such that} \\ l(\bar{x}, \tilde{y}(\bar{x})) = \inf_{x \in X} l(x, \tilde{y}(x)) \end{cases} \quad (6.3)$$

Definition 6.1

Any solution $\bar{x} \in X$ to the problem \mathcal{S} is called a Stackelberg strategy for the leader.

In the game Γ , once the leader chooses a Stackelberg strategy $\bar{x} \in X$, the optimal choice the follower will take is $\bar{y} = \tilde{y}(\bar{x})$. Such a pair is called also a

Stackelberg equilibrium. Several applications of this problem can be found in the literature: for example in Economics ([93], [94]), in Telecommunications ([95]), in Transportation ([96]). The concept of Stackelberg strategy describes situations where there is asymmetric information between the involved agents: there is one agent - the leader - who knows the best reply of the opponent - the follower - and it has been extended to the case of multiple followers situations, where the followers react to the leader's decision by solving a Nash equilibrium problem (see, for example, [95], [97], [98], [99]).

6.2 A novel entropy-based model

In this section we introduce a process of optimization under uncertainty as a conflict between players. In presence of incomplete informations about the aleatory uncertainties or when we consider epistemic uncertainties, it is still possible to identify a probability distribution which represents the least biased estimate possible on the given information. A concept which comes from information theory, the entropy, will be adopted to obtain this estimate. Hence we can imagine a Stackelberg game where the leader handles the optimization problem in presence of aleatory uncertainties with one of the methodologies described in chapter 3, while the follower provides this particular least biased distribution of the given information about uncertainties.

6.2.1 The maximum entropy principle

Historically, the first method of assigning probabilities to the outcomes of a random event was this: when there is no reason to do otherwise, assign all outcomes equal probability. This is called the principle of insufficient reason, or principle of indifference. It corresponds to a decision to use a uniform probability distribution. How do we select a probability distribution to use if we do know something about the non-uniformity of the outcomes? There is an extension of the principle of insufficient reason which suggests what to do. It is the principle of maximum entropy.

Principle 6.1

The maximum entropy estimate is the least biased estimate possible on the given information; i.e., it is the maximally noncommittal with regard to the missing information [100].

6.2.2 Entropy of a probability distribution

For a continuous probability density function $p(\xi)$ on Ω , its entropy is defined to

$$H(p) = - \int_{\Omega} p(\xi) \log(p(\xi)) d\xi \quad (6.4)$$

If the variable ξ assumes the discrete values $(\xi_1, \xi_2, \dots, \xi_N)$, our partial understanding of the process which determine the value of ξ can be represented by assigning corresponding probabilities (p_1, p_2, \dots, p_N) . [100] Hence for a

discrete probability distribution the entropy of p is defined to be

$$H(p) = - \sum_1^N p_i \log(p_i) \quad (6.5)$$

This definition of entropy, introduced by Shannon [101], resembles a formula for a thermodynamic notion of entropy. Physically, systems are expected to evolve into states with higher entropy as they approach equilibrium. In our probabilistic context, $H(p)$ is viewed as a measure of the information carried by p , with higher entropy corresponding to less information (more uncertainty, or more of a lack of information).

6.2.3 Optimization under Uncertainty

In chapter 3 it was introduced an objective function $f(z, \xi)$ where $z \in Z$ represented a design variable and $\xi \in \Omega$ a random variable. Let's now assume a given set $(\xi_1, \xi_2, \dots, \xi_i, \dots, \xi_N)$ of values assumed by the uncertain variable, the corresponding discrete Probability Density Function, \mathcal{P}^ξ , is defined by the set $p^\xi = (p_1^\xi, p_2^\xi, \dots, p_i^\xi, \dots, p_N^\xi) \in [0, 1]^N$ such that

$$\begin{cases} p_i^\xi = \mathcal{P}^\xi(\xi_i), & i \in [1, \dots, N] \\ \sum_1^N p_i^\xi = 1 \end{cases} \quad (6.6)$$

Therefore it is possible to define the leader's and the follower's problem of the proposed Stackelberg game.

6.2.4 The follower

Let's assume that we want to find the least biased estimate possible of the objective function, $f(z, p^\xi)$, given the optimal p^ξ that solves a maximum

entropy problem. Given $z \in Z$, the strategy of the follower represents the discrete PDF, p^ξ , of the given uncertainty, ξ . The payoff of the follower is represented by the entropy of the objective function, that can be written in the following form

$$H(z, p^\xi) = - \sum_1^N p_i^f \log(p_i^f) \quad (6.7)$$

considering

$$\begin{cases} f_i = f(z, \xi_i) \\ (f, p^f) = \Xi_z(\xi, p^\xi) \end{cases} \quad (6.8)$$

where the operator $\Xi_z : \{\xi, p^\xi\} \in \Omega \times R^N \rightarrow \{f, p^f\} \in R^N \times R^N$ is the propagation process of the input uncertainty in the quantity of interest (QoI) and it could be represented by different methodologies (e.g. Monte Carlo). It is important to notice that the operator Ξ_z depends on the design variable chosen by the leader, z .

6.2.5 The leader

Let's assume that the leader of the Stackelberg game considers as payoff a deterministic attribute, Φ , of the objective function, $f(\cdot)$, which is probabilistic in nature. Hence the leader looks for z in the design set Z , minimizing $\Phi(f(\cdot))$ given the optimal input PDF of the uncertainty found by the follower that is supposed to be unique $\forall z \in Z$. This unique solution is denoted as $\tilde{p}^\xi(z)$. As an example the deterministic attribute of the objective function could be the mean of the objective function itself, hence

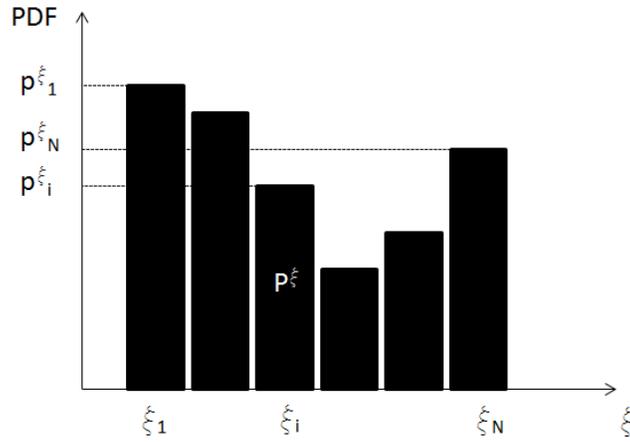


Figure 6.1: Probability distribution function of the input uncertainty

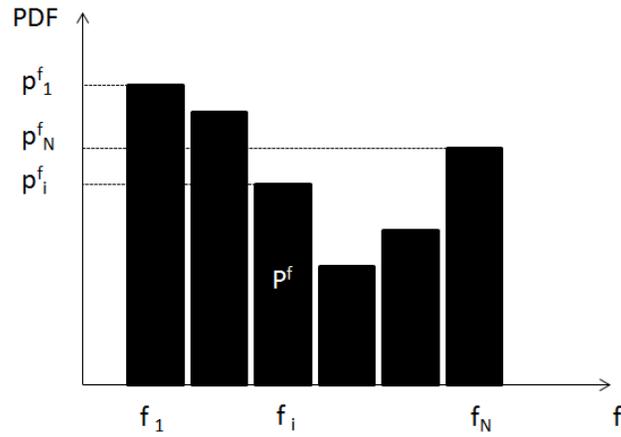


Figure 6.2: Probability distribution function of the output

$$\Phi(f(z, p^\xi)) = \mu(f(z, p^\xi)) = \sum_1^N p_i^f f(z, x_i) \quad (6.9)$$

6.2.6 The game

Let $z \in Z$ be the leader's announced choice. For any $z \in Z$, the follower solves the following lower level problem, $\mathcal{P}(z)$

$$\begin{cases} \text{find } \bar{p}^\xi \in R^N \text{ such that} \\ H(z, \bar{p}^\xi) = \inf_{p^\xi \in R^N} -H(z, p^\xi) \end{cases} \quad (6.10)$$

The solution to this problem is supposed to be unique and denoted by $\tilde{p}^\xi(z)$. The function $z \in Z \rightarrow \tilde{p}^\xi(z) \in R^N$ is called follower's best reply or best response. The leader knows this best reply and optimizes his own updated cost function just considering the follower's response and solves the following upper level problem, \mathcal{S}

$$\begin{cases} \text{find } \bar{z} \in Z \text{ such that} \\ \Phi(f(\bar{z}, \tilde{p}^\xi(\bar{z}))) = \inf_{z \in Z} \Phi(f(z, \tilde{p}^\xi(z))) \end{cases} \quad (6.11)$$

A summary of the entropy-based game is reported in Table 6.1.

Table 6.1: The entropy-based game

	strategy	payoff
leader	z	$\Phi(f(z, p^\xi))$
follower	p^ξ	$-H(z, p^\xi)$

6.3 The Stackelberg-Nash model

We deal with a 2+n player game Γ , where two players are the leaders and the rest of them are followers in a two-level Stackelberg game[102]. Any player is assumed to minimize his own payoff function called cost function.

The followers, as well as the leaders, act themselves in a noncooperative way and play a Nash equilibrium game[102]. The leaders take into account the followers Nash equilibrium, that we assume to be unique, and solve a Nash equilibrium problem in a backward induction scheme.

Let $X_1, X_2, Y_1, \dots, Y_n$ be the leaders' and the followers' strategy sets, respectively. Let $l_1, l_2, f_1, \dots, f_n$ be real valued functions defined on $X_1 \times X_2 \times Y_1 \times \dots \times Y_n$ representing the leaders' and the followers' cost functions. The leaders announce $(x_1, x_2) \in X_1 \times X_2$ and the lower level problem, i.e. the followers problem, is a n person non-cooperative game $\Gamma_n(x_1, x_2) = \langle n; Y_1, \dots, Y_n; f_1, \dots, f_n \rangle$.

For each $(x_1, x_2) \in X_1 \times X_2$, that is the leaders' decisions, the followers solve the following *lower level Nash equilibrium problem* $\mathcal{N}(x_1, x_2)$:

$$\left\{ \begin{array}{l} \text{find } (\bar{y}_1, \dots, \bar{y}_n) \in Y_1 \times \dots \times Y_n \text{ such that} \\ f_1(x_1, x_2, \bar{y}_1, \dots, \bar{y}_n) = \inf_{y_1 \in Y_1} f_1(x_1, x_2, y_1, \bar{y}_2, \dots, \bar{y}_n) \\ \dots \\ f_n(x_1, x_2, \bar{y}_1, \dots, \bar{y}_n) = \inf_{y_n \in Y_n} f_n(x_1, x_2, \bar{y}_1, \dots, \bar{y}_{n-1}, y_n). \end{array} \right. \quad (6.12)$$

Let $(\tilde{y}_1(x_1, x_2), \dots, \tilde{y}_n(x_1, x_2)) \in Y_1 \times \dots \times Y_n$ be the unique solution of the problem $\mathcal{N}(x_1, x_2)$. The leaders take into account the followers' decision and solve a Nash equilibrium problem corresponding to a 2 persons non-cooperative game $\Gamma_2 = \langle 2; X_1, X_2; l_1, l_2 \rangle$. They compute a solution of the following *upper level Nash equilibrium problem* \mathcal{SN} :

$$\left\{ \begin{array}{l} \text{find } (\bar{x}_1, \bar{x}_2) \in X_1 \times X_2 \text{ such that} \\ l_1(\bar{x}_1, \bar{x}_2, \tilde{y}_1(\bar{x}_1, \bar{x}_2), \dots, \tilde{y}_n(\bar{x}_1, \bar{x}_2)) = \inf_{x_1 \in X_1} l_1(x_1, \bar{x}_2, \tilde{y}_1(x_1, \bar{x}_2), \dots, \tilde{y}_n(x_1, \bar{x}_2)) \\ l_2(\bar{x}_1, \bar{x}_2, \tilde{y}_1(\bar{x}_1, \bar{x}_2), \dots, \tilde{y}_n(\bar{x}_1, \bar{x}_2)) = \inf_{x_2 \in X_2} l_2(\bar{x}_1, x_2, \tilde{y}_1(\bar{x}_1, x_2), \dots, \tilde{y}_n(\bar{x}_1, x_2)) \end{array} \right. \quad (6.13)$$

Definition 6.2

Any solution $(\bar{x}_1, \bar{x}_2) \in X_1 \times X_2$ to the problem \mathcal{SN} is called a Stackelberg-Nash strategy, while any vector $(\bar{x}_1, \bar{x}_2, \tilde{y}_1(\bar{x}_1, \bar{x}_2), \dots, \tilde{y}_n(\bar{x}_1, \bar{x}_2)) \in X_1 \times X_2 \times Y_1 \times \dots \times Y_n$ is called a Stackelberg-Nash equilibrium of the game Γ .

This model has been intensively studied and used in different applicative contexts [111, 104, 96, 99].

To ensure the uniqueness of the Nash equilibrium of the lower level, the well known dominant diagonal condition [105] for smooth games can be applied. Recall that special classes of games admit a unique Nash equilibrium point, provided suitable assumptions on the data: for example quadratic games[102] or some oligopolistic games[106].

6.4 Algorithmic Game Theory

Several papers dealing with algorithmic Game Theory can be found in the literature. Particularly, as done in [63], [107], [108], [81], some numerical results have been obtained for game-theoretical models by using the genetic algorithmic approach. A computational procedure to compute Stackelberg strategies based on a GA has been given in [109]. The proposed algorithm

considers the uniqueness of the follower's best response and the classical cross-over tournament procedures.

In this section a GA algorithm to compute the Nash equilibrium, the Stackelberg strategy and the Stackelberg-Nash strategy is presented . In Section 6.5.2 an algorithm to evaluate Stackelberg strategy in the case where the follower's response is not unique is presented. We assume an optimistic leader's behavior and consider the so-called strong Stackelberg solution: the leader can force the follower's choice in the best reply set. The multi-modal approach presented in [98] will be used to manage the follower's multiple response, i.e. the multiplicity of solutions to the problem $\mathcal{P}(x)$, $\forall x \in X$. The algorithm proposed considers also the possibility of multiple leader's solutions: the multi-modal approach is again used in the upper level problem.

6.4.1 GA for a Nash equilibrium problem

In this subsection we illustrate the genetic algorithm to compute a Nash equilibrium. For simplicity we consider a 2 person game. Let Y_1, Y_2 be compact subsets of metric spaces that are the players' strategy sets. Let f_1, f_2 be two real valued functions defined on $Y_1 \times Y_2$ representing the players' cost functions. A Nash equilibrium problem is solved by the two players.

Let $\vec{s} = u, v$ be the string (or individual, or chromosome) representing the potential solution for a 2 person Nash problem.

Then u denotes the subset of variables handled by player 1, belonging to a metric space Y_1 , and optimized under an objective function always denoted by f_1 . Similarly v indicates the subset of variables handled by player 2,

belonging to metric space Y_2 , and optimized along another objective function denoted by f_2 . Thus, as advocated by Nash theory[91], player 1 optimizes the chromosome with respect to the first objective function by modifying u while v is fixed by player 2; symmetrically, player 2 optimizes the chromosome with respect to the second criterion by modifying v , while u is fixed by player 1.

Let u^{k-1}, v^{k-1} be the best value found by player 1, player 2 respectively at generation $k - 1$. At next generation k , player 1 optimizes u^k using v^{k-1} in order to build its chromosome (now $\vec{s} = u^k, v^{k-1}$). At the same time player 2 optimizes v^k using u^{k-1} to evaluate his chromosome (in this case $\vec{s} = u^{k-1}, v^k$).

The algorithm is organized in several steps (as shown in Figure 6.9) that consist of:

1. Creating two different random populations, one for each player only at the first generation. Player 1's optimization task is performed by population 1 and vice versa;
2. The classification is made on the basis of the evaluation of a fitness function, typical of GAs, that counts the results of matches between each individual of population 1 with all individuals of population 2, scoring 1 or -1 for a win or loss, and 0 for a draw.

$$\begin{cases} \text{if } f_1(u_i^k, v^{k-1}) > f_1(u^{k-1}, v_i^k), \text{fitness}_1 = 1 \\ \text{if } f_1(u_i^k, v^{k-1}) < f_1(u^{k-1}, v_i^k), \text{fitness}_1 = -1 \\ \text{if } f_1(u_i^k, v^{k-1}) = f_1(u^{k-1}, v_i^k), \text{fitness}_1 = 0 \end{cases}$$

The same is made for player 2:

$$\begin{cases} \text{if } f_2(u_i^{k-1}, v^{k-1}) < f_2(u^{k-1}, v_i^k), \textit{fitness}_2 = 1 \\ \text{if } f_2(u_i^k, v^{k-1}) > f_2(u^{k-1}, v_i^k), \textit{fitness}_2 = -1 \\ \text{if } f_2(u_i^k, v^{k-1}) = f_2(u^{k-1}, v_i^k), \textit{fitness}_2 = 0 \end{cases}$$

In this way a simple sorting criterion could be established. For equal fitness value individual are sorted on objective function f_1 for population 1 (player 1) and on objective function f_2 for player 2;

3. A mating pool for parent chromosome is generated and common GA techniques as crossover and mutation are performed on each player population. A second sorting procedure is needed after this evolution process.
4. At the end of k -th generation optimization procedure player 1 communicates his own best value u^k to player 2 who will use it at generation $k + 1$ to generate its entire chromosome with a unique value for its first part, i.e. the one depending on player 1, while on the second part comes from common GAs crossover and mutation procedure. Conversely, player 2 communicates its own best value v^k to player 1 who will use it at generation $k + 1$, generating a population with a unique value for the second part of chromosome, i.e. the one depending on player 2;
5. A Nash equilibrium is reached when a generation number limit is reached or an exit criterion on the difference between objective functions from two subsequent generations is met within the steps 2-4.

This kind of structure for the algorithm is similar to those used by other researchers, with a major emphasis on fitness function consistency[108, 81].

6.4.2 GA for a Stackelberg-Nash equilibrium problem

Here the algorithm is presented for a 2+n person hierarchical game or Stackelberg-Nash game with two leaders and n followers. Let $X_1, X_2, Y_1, \dots, Y_n$ be compact subsets of metric spaces that are the players' strategy sets. Let $l_1, l_2, f_1, \dots, f_n$ be real valued functions representing the players' cost functions. The algorithm is organized in several steps (as shown in Figure 6.10) that consist of:

1. The initial population is provided with a random seeding in the subset of a metric space, $X_1 \times X_2$, i.e. that of the leaders. This is made only at the first generation;
2. For each individual (or chromosome) of the leaders population, say $u_{1,i}, u_{2,i}$, a random population for each follower player is generated, i.e. providing $v_{1,i}, \dots, v_{n,i}$;
3. At this step a typical Nash equilibrium search for the followers is made until is reached a limit in generation number in a similar manner to that shown in subsection 6.4.1;
4. As a result of the previous step we have determined the followers player best reply, i.e. the unique Nash equilibrium. This is now ready to be passed to the leader players in order to evaluate their Nash equilibrium by sorting the population under fitness criterion;

5. The results of sorting procedure is a classification of leaders population with best individual denoted by u_1^k for leader 1 and u_2^k for leader 2. Now a common crossover and mutation operations on the leaders population are performed only on u_1 for leader 1 with a unique value of u_2 equal to u_2^k for each individual in leader 1 population, such that the chromosome could be represented as $\vec{s} = u_1, u_2^k, v_1, \dots, v_n$. The same operation is made on leader 2 population with switched variables;
6. This new population need to be sorted again by fitness criterion in order to give the u_1^{k+1}, u_2^{k+1} from which starts the following generation;
7. A Stackelberg-Nash solution is determined when a generation limit for leader players population is reached or an exit criterion on difference between objective functions from two subsequent generations is met within step 2-6.

6.4.3 Algorithm validation

For the algorithm validation a test case is considered with two leaders and two followers. Let $X_1 = X_2 = Y_1 = Y_2 = [0, 1]$ be the strategy sets of the two leaders and the two followers. The players' cost functions are, respectively, the following:

$$\begin{cases} l_1(x_1, x_2, y_1, y_2) = 2x_1y_2 - 2x_2y_1 - x_1x_2 \\ l_2(x_1, x_2, y_1, y_2) = 2(y_1 + y_2)(x_1 + x_2) - 4x_1x_2 - x_2 \\ f_1(x_1, x_2, y_1, y_2) = (y_2 - y_1 - x_1)^2 + (2y_1 - x_2 - x_1)^2 \\ f_2(x_1, x_2, y_1, y_2) = 4(y_1 - y_2 - x_1)^2 + (2y_2 - x_2 - 2x_1)^2 \end{cases} \quad (6.14)$$

In this case $(\tilde{y}_1(x_1, x_2), \tilde{y}_2(x_1, x_2)) = (x_2/2 + 2x_1/9, x_2/2 + x_1/9)$ for any $(x_1, x_2) \in [0, 1]^2$ and

$$l_1(x_1, x_2, \tilde{y}_1(x_1, x_2), \tilde{y}_2(x_1, x_2)) = 2x_1^2/9 - x_2^2 - 4/9x_1x_2$$

$$l_2(x_1, x_2, \tilde{y}_1(x_1, x_2), \tilde{y}_2(x_1, x_2)) = 2x_1^2/3 + 2x_2^2 - 4/3x_1x_2 - x_2$$

The analytical Stackelberg-Nash solution is

$$(\bar{x}_1, \bar{x}_2, \bar{y}_1, \bar{y}_2) = \left(3/8, 3/8, 13/48, 11/48 \right)$$

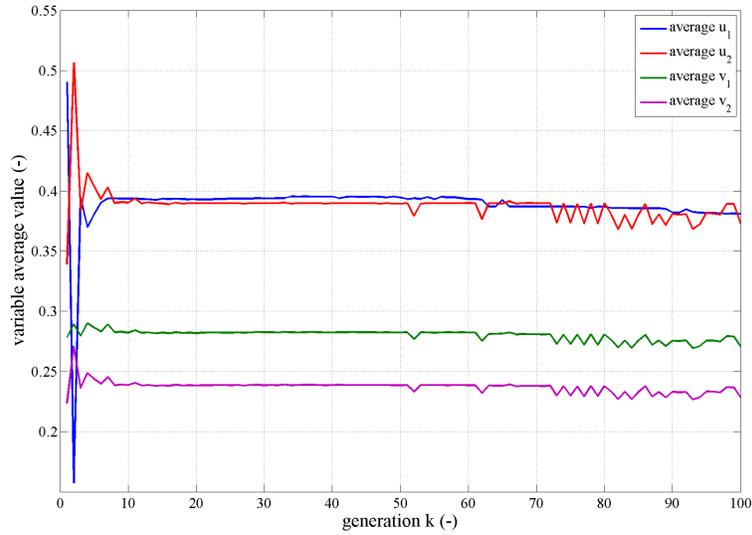
where $\bar{y}_i = \tilde{y}_i(\bar{x}_1, \bar{x}_2)$, $i = 1, 2$, while the numerical solution is

$$(\bar{u}_1, \bar{u}_2, \bar{v}_1, \bar{v}_2) = \left(0.3962, 0.3828, 0.2745, 0.2319 \right).$$

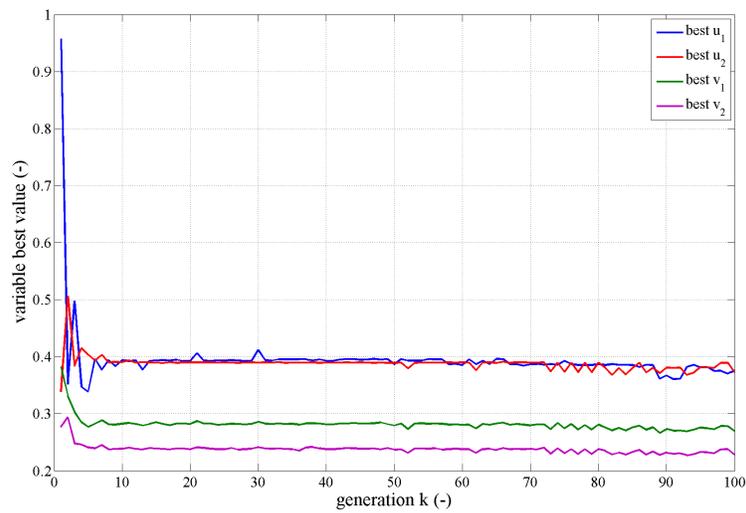
In figure 6.3 are shown, respectively, the variables average value and the variables best value through the generation number k . These numerical solutions are indicated with u_1, u_2 for the leader's variables and v_1, v_2 for the follower's variables.

In the above test case the characteristics of GAs algorithm could be summarized in Table 6.2.

Finally we would like to point out the sensitivity of our algorithm with respect to the number of players considered in the game. In figure 6.4 the non dimensional times needed to complete each generation k (up to 20), for several leader follower arrangements, are shown. In particular, we refer all the times to the $1l + 1f$ situation, indicating with a dashed line the average on generations number. We have limited this sensitivity analysis to a 2 leaders - 2 followers problem, that is the situation faced during the test case.



(a) variable average value



(b) variable best value

Figure 6.3: Average and best variables value through generation

Table 6.2: GA details

Parameter	Value
Population size [-]	50
Crossover fraction [-]	0.90
Mutation fraction [-]	0.10
Parent sorting	Tournament between couple
Mating Pool [%]	50
Elitism	no
Crossover mode	Simulated Binary Crossover (SBX)
Mutation mode	Polynomial
Generation Limit on Leader	50
Tolerance on objective functions	10^{-5}

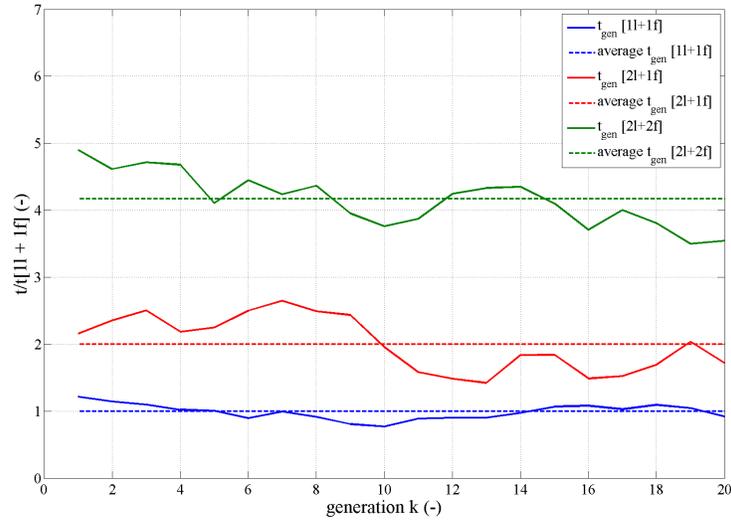


Figure 6.4: Non dimensional time ratio through generation

6.4.4 Applications to real life problems

The methodology and the computational algorithm developed in this chapter could be very useful in many engineering problems, like optimization

tool alternative to classical gradient methods (see also [107], [108], [81]). In particular the hierarchical Stackelberg-Nash model would be very attractive for real engineering problems like the optimization of the position of a multi element airfoil or wing: in these cases important targets for all the movable surfaces that constitute the airfoil (like slats and flaps) or wing (aileron, rudder, elevator, etc.) are present. This kind of methodology (leader-follower) would avoid the traditional troubles faced in multi-objective optimization problem where a scalarization approach is needed to weight every request (objective function) to form a “new” scalar function to optimize. Among others, it would be also very useful in a classic aircraft preliminary design where some specifications appear to be dominant (leader) with respect to others (followers) in determining aircraft shape and dimensions.

Applications in other contexts must be mentioned too: the hierarchical Stackelberg-Nash model has been used, for example, in oligopolies as in [99] or in Transportation problems as in [96].

6.5 The non-uniqueness

As it happens in many cases, the lower level problem may have more than one solution for at least one $x \in X$. Let us define for any $x \in X$ the set $R(x)$ of the solution to the problem $\mathcal{P}(x)$. The correspondence defined on X and valued in Y mapping to any $x \in X$ the subset $R(x) \subseteq Y$ of all possible solutions to the problem $\mathcal{P}(x)$, is called follower’s best reply or best response. In this case the best reply is a multi-valued function and the upper level problem has to be formulated depending on the leader’s behavior. The

leader has to optimize the updated cost function, but he does not know what is the follower's choice in the set $R(x)$. So, a possible approach is that the leader suppose that the follower's choice is the best for himself and solve the following upper level problem, \mathcal{S}^s

$$\begin{cases} \text{find } \bar{x} \in X \text{ such that} \\ \inf_{y \in R(\bar{x})} l(\bar{x}, y) = \inf_{x \in X} \inf_{y \in R(x)} l(x, y) \end{cases} \quad (6.15)$$

Definition 5.3

Any solution \bar{x} to the problem \mathcal{S}^s is called a strong Stackelberg strategy for the leader.

This solution concept corresponds to an optimistic leader's point of view ([110], [111]) and it is also known as generalized Stackelberg strategy ([112]). This is not the only possibility to define a Stackelberg strategy in the case of multiple follower's best reply, in the concluding section we address to some other definition. In the following we deal with strong Stackelberg strategies.

Remark 5.1

Let us define for any $x \in X$ the following marginal function

$$v(x) = \inf_{y \in R(x)} l(x, y).$$

Note that if $\bar{x} \in X$ is a strong Stackelberg strategy for the leader, any pair (\bar{x}, y) with $y \in R(\bar{x})$ is a possible exit for the game. Any choice $y \in R(\bar{x})$ is equivalent for the follower, since $R(\bar{x}) = \operatorname{argmin}_{y \in Y} f(\bar{x}, y)$. For the leader, the situation is different and if there exists a $y^b \in R(\bar{x})$ such that $v(\bar{x}) = l(\bar{x}, y^b)$, the choice (\bar{x}, y^b) will be the best for him.

Definition 5.4

Any pair $(\bar{x}, \bar{y}) \in X \times Y$ such that \bar{x} is a solution to the problem \mathcal{S}^s and $\bar{y} \in \operatorname{argmin}_{y \in R(\bar{x})} l(\bar{x}, y)$ is called a strong hierarchical equilibrium.

In the following we will assume that:

- (i) X, Y are compact subsets of metric spaces;
- (ii) l, f are continuous functions on $X \times Y$.

Under assumptions (i), (ii) the marginal function $v(x)$ turns out to be a continuous function on X , then there exists at least a strong hierarchical equilibrium ([113]).

6.5.1 Multi-modal optimization

The multi-modal optimization is a technique used to find all relative maxima or minima of an objective function. A GA procedure is implementable to reach this aim by introducing a so-called “*sharing function*”, i.e. a penalty function that grows as well as increases the density of points close to relative maximum or minimum for the objective function. For each individual of the population we compute a relative distance based e.g. on chromosome s for a simple case of two design variables

$$d_{i,j} = \sqrt{\sum_{k=1}^r (s_{i,k} - s_{j,k})^2}, \quad \forall i, j = 1, \dots, n \quad (6.16)$$

where r is the number of design variables (or genes, at least one for each player as written for s above), n the population size and $d_{i,j}$ the distance between individuals i and j . The “*sharing function*” is defined as follows

$$p_{i,j} = 1 + \log\left(\frac{d_{i,j}}{d_{min}}\right), \begin{cases} p_{i,j} \geq 1 \Rightarrow p = 1 \\ p_{i,j} \leq 0.01 \Rightarrow p = 0.01 \end{cases} \quad (6.17)$$

where d_{min} is the minimum distance allowed between different individual in the same population and is based usually on domain extent. At this point a multi-modal fitness function for each individual i in population can be evaluated from previous fitness function

$$fitness_i^{mm} = fitness_i \cdot \prod_{j=1}^n p_{i,j} \quad (6.18)$$

We will use this function to approach the multi-modal case in the lower level problem as well as in the upper one.

6.5.2 Algorithm modifications for the non-uniqueness

We present here the algorithm for a two player Stackelberg equilibrium game with non unique follower's best response. Let X, Y be compact subsets denoted by players' strategy sets. Let l, f be two real valued functions defined on $X \times Y$ representing the players' cost functions.

The algorithm is organized in several steps that consist of:

1. The initial population is provided with a random seeding in the subset of a metric space, X , i.e. that of the leader. This is made only at the first generation;
2. For each individual (or chromosome) of the leader population, say u_i , a random population for the follower player is generated, i.e. providing $v_{1,i}, \dots, v_{n,i}$;

3. At this step a best reply research for the follower is made until is reached a limit in generation number;
4. Among the best replies a strong Stackelberg solution is chosen by the leader in order to evaluate his payoff by sorting the population under fitness criterion;
5. The results of sorting procedure is a classification of leaders population with best individual denoted by u^k . Now a common crossover and mutation operation on the leaders population is performed;
6. A strong Stackelberg solution is determined when a generation limit for leader players population is reached or an exit criterion on difference between objective functions from two subsequent generations is met within steps 2-5.

This kind of structure for the algorithm is similar to those used by other researchers, with a major emphasis on fitness function consistency ([81]).

The algorithm considers all the possible solutions of the lower level problem (step 3), then selects between all these values for the follower only those minimizing the leader's cost function (steps 4-5) then there is an optimization problem solved for the leader (step 6). Assumptions (i), (ii) give the existence of at least a strong hierarchical equilibrium, and because of the algorithm construction the follower chooses at every step his best reply, the leader ends up to any value in

$$\operatorname{argmin}_{x \in X} \min_{y \in R(x)} l(x, y).$$

The convergence result is given in the following proposition.

Proposition 6.1

Under assumptions (i), (ii) the GA algorithm converges to strong hierarchical equilibrium pairs $(\bar{x}, \bar{y}) \in X \times Y$.

6.5.3 A first numerical example

Let us consider the 2-person game with strategy sets $X = Y = [0, 1]$ and cost functions

$$\begin{cases} l(x, y) = -xy \\ f(x, y) = -y(y - x) \end{cases} \quad (6.19)$$

We have

$$R(x) = \begin{cases} 1 & \text{if } x < 1 \\ \{0, 1\} & \text{if } x = 1 \end{cases} \quad (6.20)$$

so the follower's best reply is not single-valued at $x = 1$. The leader, supposed to be optimistic, has to minimize the function

$$\inf_{y \in R(x)} -xy = -x$$

and a strong Stackelberg solution is $\bar{x} = 1$. Let us remark that the follower may react to this solution with $y = 0$ or $y = 1$, and since $f(1, 0) = f(1, 1)$ for him it is equivalent. Not the same happens for the leader because $l(1, 0) = 0$ and $l(1, 1) = -1$ that is better for him.

For the algorithm validation we consider the parameters specified in Table 6.2 with a value of $d_{min} = 0.2$ for multi-modal optimization.

Our algorithmic procedure gives the following numerical results

$$\hat{x} = 1.000, \quad \hat{y} = 1.000.$$

We point out that the algorithm selects the best result from the leader point of view, as discussed in Remark 5.1. In this example, the multi-modal approach as been used in the lower level problem.

6.5.4 A second numerical example

Let us consider the 2-person game with strategy sets $X = Y = [-1, 1]$ and cost functions

$$\begin{cases} l(x, y) = -xy \\ f(x, y) = (y^2 - x^2)^2 \end{cases} \quad (6.21)$$

We have

$$R(x) = \{x, -x\} \quad (6.22)$$

so the follower's best reply is not single-valued at any $x \neq 0$. The leader, supposed to be optimistic, has to minimize the function

$$\inf_{y \in R(x)} -xy = -x^2$$

and we find two strong Stackelberg strategies $\bar{x}_1 = -1$ and $\bar{x}_2 = 1$. Analytically, we have

$$\bar{x}_1 = -1, \quad \bar{y}_1 \in \{-1, 1\}$$

$$\bar{x}_2 = 1, \quad \bar{y}_2 \in \{-1, 1\}$$

$$l(-1, -1) = l(1, 1) = -1.$$

By considering the multi-modal approach in the lower level problem as well as in the upper one, our algorithmic procedure gives the following numerical results

$$\hat{x}_1 = -0.999, \quad \hat{y}_1 = -0.968$$

$$\hat{x}_2 = 1.000, \quad \hat{y}_2 = 1.000.$$

Remark that in this example, the selection $y^b = x$ in the best reply set gives the best value for the leader.

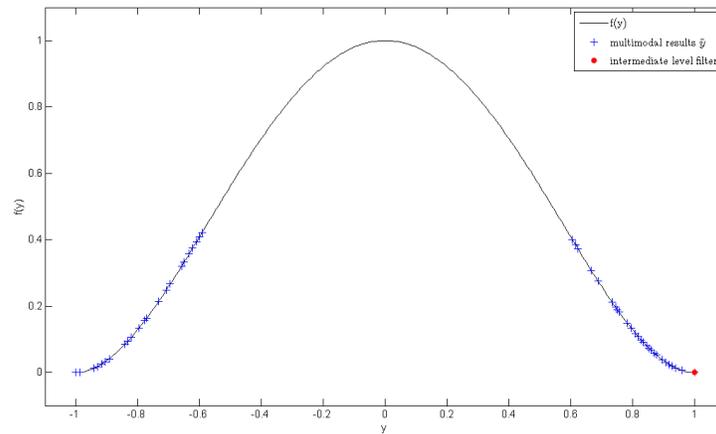


Figure 6.5: Follower's objective function towards generations with multimodal approach

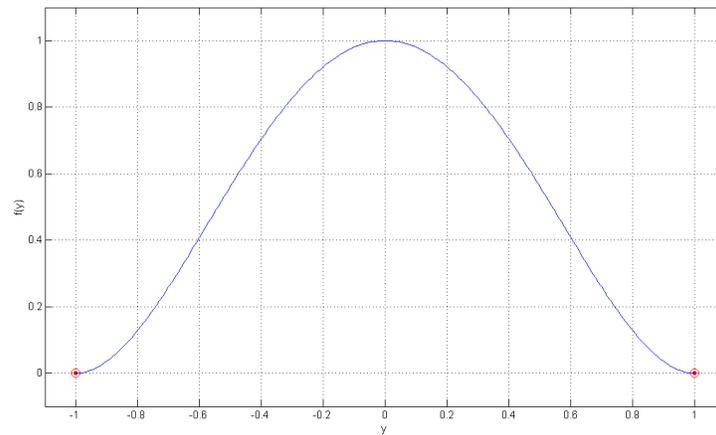


Figure 6.6: Follower's objective function towards generations with an intermediate filter operation on multi-modal approach

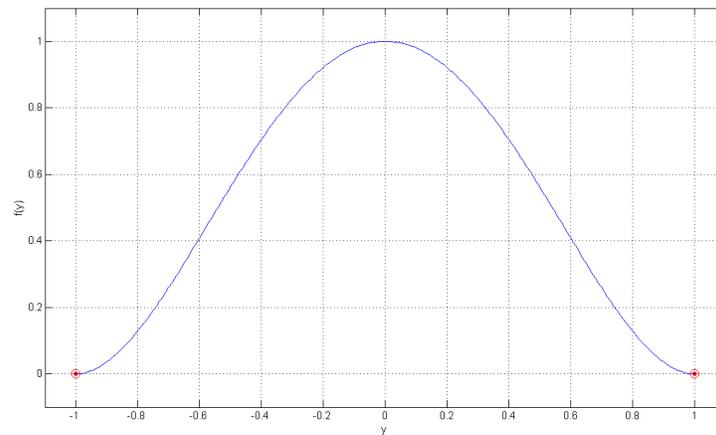


Figure 6.7: Follower's objective function towards generations after filtering

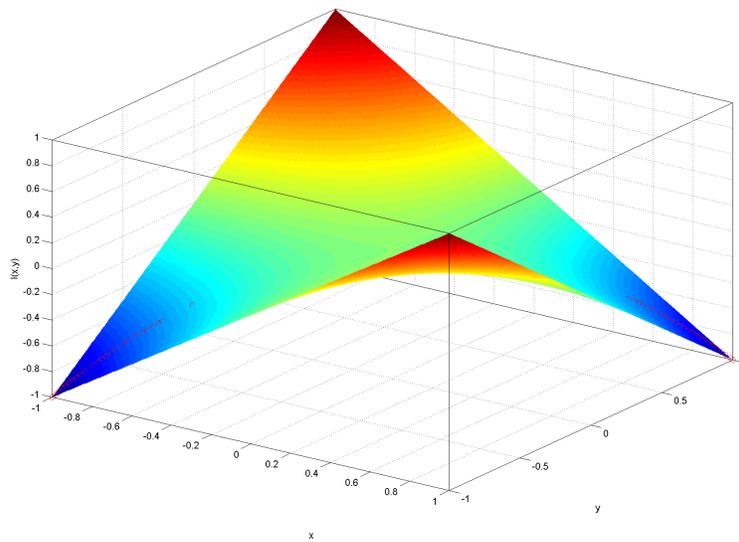


Figure 6.8: Leader's objective function

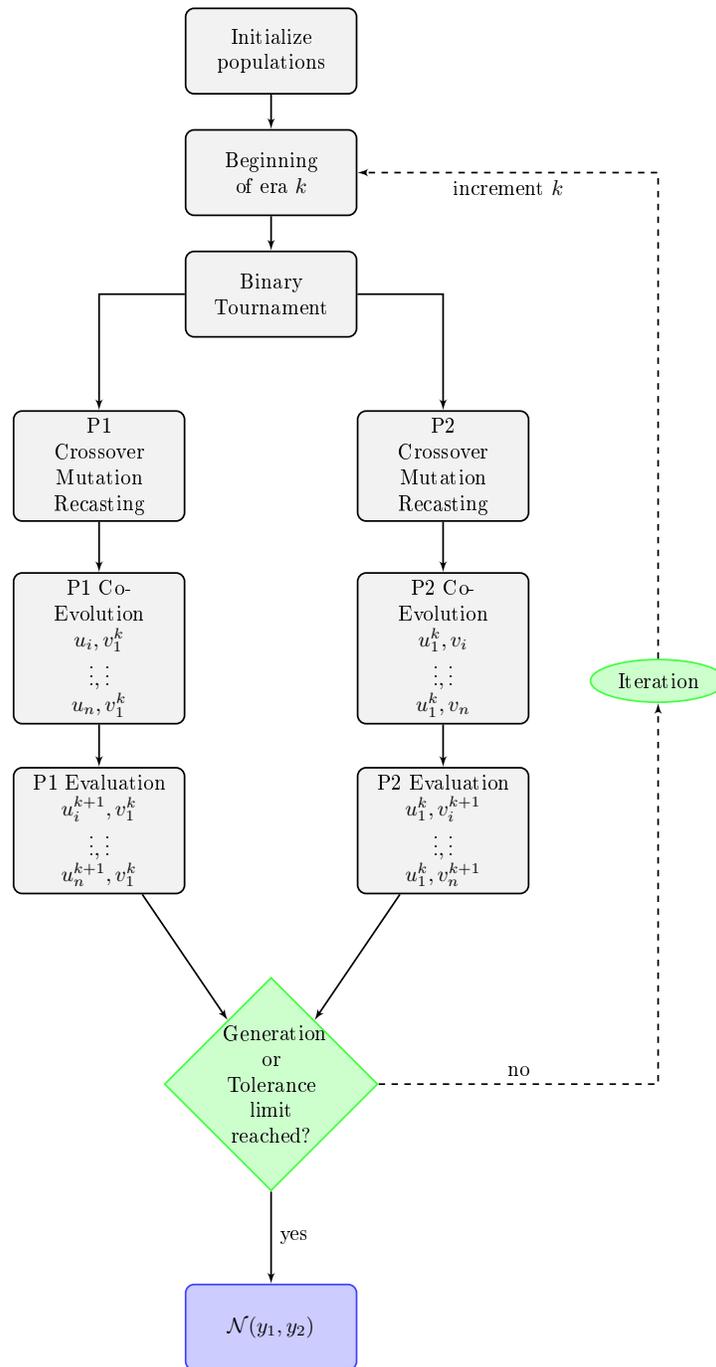


Figure 6.9: Nash equilibrium GA structure

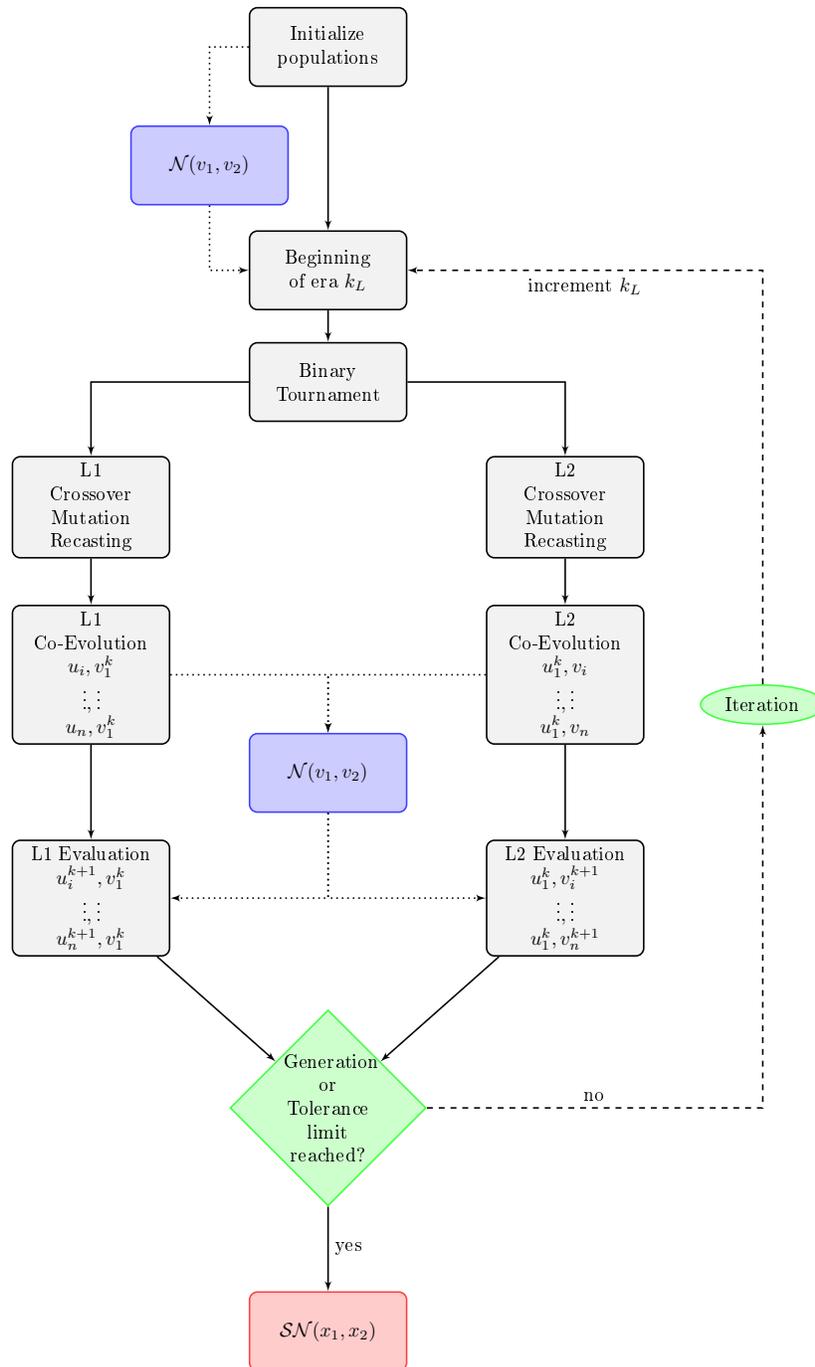


Figure 6.10: Stackelberg-Nash equilibrium GA structure

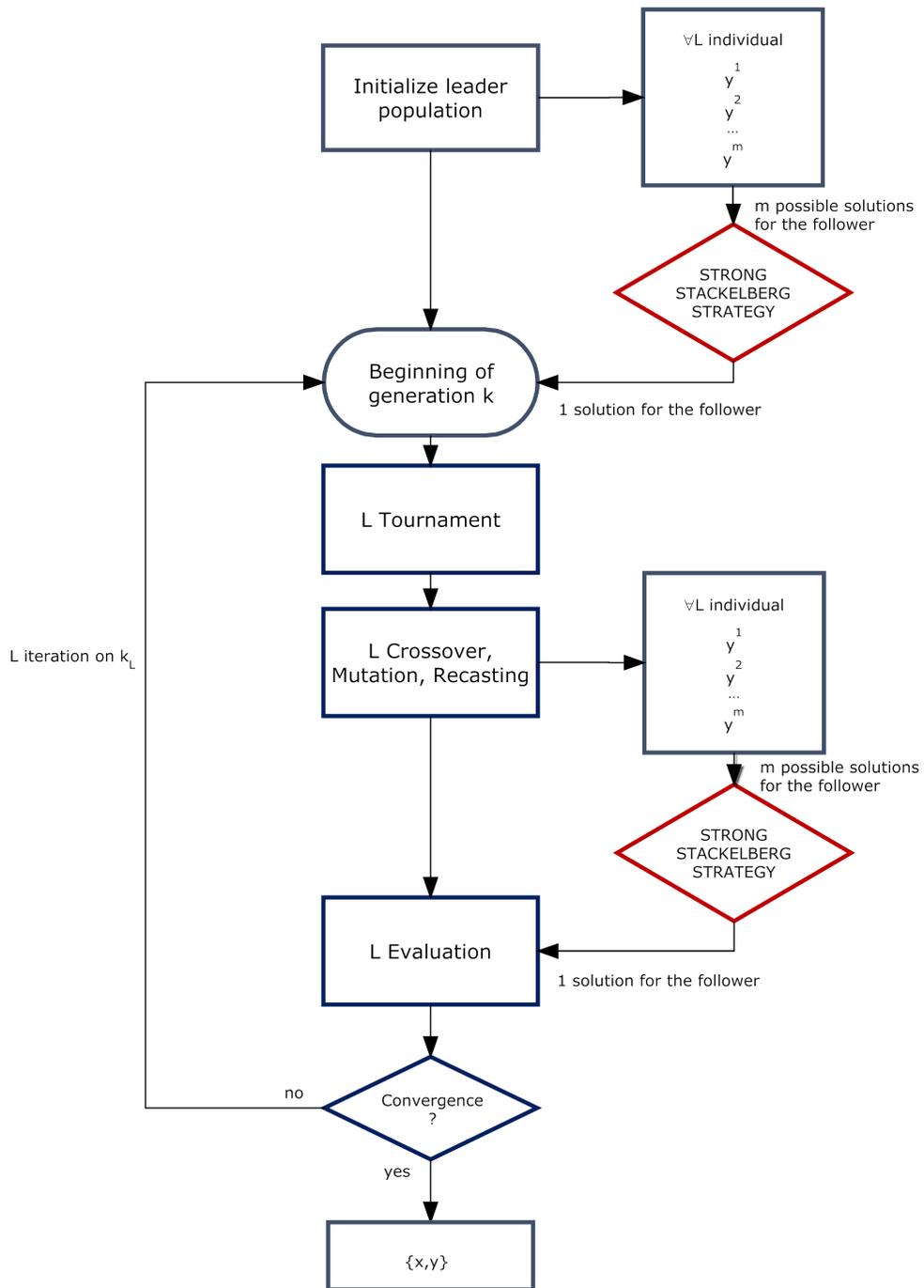


Figure 6.11: Stackelberg-GAs algorithm structure with non-uniqueness

Chapter 7

Conclusions

The primary goals of this work were to develop algorithms for efficient uncertainty quantification and optimization under uncertainty and to use them in industrial applications. Both goals have been achieved since the effectiveness of the Simplex Stochastic Collocation for analysis under uncertainty has been proven and a novel theoretical formulation that allows a tight coupling with the optimization procedure has been presented and assessed. Indeed it has been shown the necessity of take in account uncertainty since the beginning of the design process.

Two industrial applications have been presented in the development of this thesis and the current framework has been used to optimize the shape of wind turbine blades and of a Formula 1 tire brake intake. The designs obtained with the proposed procedure appeared to be more stable in the presence of uncertainty than their deterministic counterpart.

Game Theory has been proven to be a possible solution in handling problems of optimization under uncertainty where a lack of knowledge about the variability of several uncertain parameters is taken in account.

We also developed *Leland*, the key-product of this research, offering to its actual users all the algorithms that have been presented in this work. The importance of having a meta-scheduler with complete fault tolerance has been proven to be a necessity in performing the extreme ensemble computations needed for optimization under uncertainty.

Four important areas of future work are the quantification of the sensitivity of the CDF-based approaches, in order to assess the impact of the quality of the CDF estimation of the objectives on the prediction of the optimal solutions. To further prove the robustness of these methodologies an assessment of the quality of the solutions with respect to the sampling technique adopted, the dimensionality of the probability space and the size of the ensemble is required.

The second area of future work is the investigation of the performance of these CDF-based approaches in problems where the solution is known to have good mathematical properties (e.g. Gaussianity) and few statistical moments are representative of the problem. In this case the comparisons with the classic techniques to obtain the probabilistic Pareto front are meaningful for the validation of the methodology.

Furthermore a third area of future work is represented by the improvements of the meta-scheduler *Leland* and its application to a great variety of problems of the industry and the research community that require an high computational efforts. The contribution of researchers from computer science disciplines is actually improving the efficiency of the algorithms proposed in this work.

CHAPTER 7. CONCLUSIONS

The last area of future research concerns the opportunities to create a unifying framework of Game Theory, uncertainty quantification and optimization. Indeed the entropy-based Stackelberg game for optimization under uncertainty will be further developed and assessed in frameworks characterized by the presence of both epistemic and aleatory uncertainties.

Bibliography

- [1] Beale, E. M. L. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society*, 17B, 173–184.
- [2] Bellman, R. E. (1957). *Dynamic programming*. Princeton, PA: Princeton University Press.
- [3] Bellman, R., Zadeh, L. A. (1970). Decision-making in a fuzzy environment. *Management Science*, 17, 141–161.
- [4] Charnes, A., Cooper, W. W. (1959). Chance-constrained programming. *Management Science*, 6, 73–79.
- [5] Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, 1, 197–206.
- [6] Tintner, G. (1955). Stochastic linear programming with applications to agricultural economics. In H. A. Antosiewicz (Ed.), *Proceedings of the Second Symposium in Linear Programming* (pp. 197–228), National Bureau of Standards, Washington, DC.
- [7] Horner, P. (1999). Planning under uncertainty. Questions & answers with George Dantzig. *OR/MS Today*, 26, 26–30.

BIBLIOGRAPHY

- [8] Green, L., Lin, H. et Khalessi, M. 2002 Probabilistic methods for uncertainty propagation applied to aircraft design. In 20th AIAA Applied Aerodynamics Conference.
- [9] Helton, J. C. et Oberkampf, W. L. 2004 Alternative representations of epistemic uncertainty. *Reliability Engineering and System safety* 85 (1), 1–10.
- [10] Zimmermann, H. 1996 *Fuzzy set theory and its applications*, 3rd edn. Kluwer Academic Publishers
- [11] Bae, H., Grandhi, R. V. et Canfield, R. A. 2004 An approximation approach for uncertainty quantification using evidence theory. *Reliability engineering and systems safety* 86 (3), 215–225.
- [12] Mourelatos, Z. P. et Zhou, J. 2004 Uncertainty quantification using evidence theory in multidisciplinary design optimization. *Reliability Engineering and System Safety* 85 (1), 281–294.
- [13] Ellison, S. L. R., Rosslein, M. et Williams, A. 2000 Quantifying uncertainty in analytical measurement. CEPIS.
- [14] Wojtkiewicz, S. F., Eldred, M. S., Field, R. V., Urbina, A. et Red-Horse, J. R. 2001 Uncertainty quantification in large computational engineering models. In 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials (SDM) Conference.
- [15] Draper, D. 1995 Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society. Series B* 57 (1), 45–97.

BIBLIOGRAPHY

- [16] Bose, D. et Wright, M. 2006 Uncertainty analysis of laminar aeroheating predictions for mars entries. In 38th AIAA Thermophysics Conference (ed. V. Alexandrov et al.), pp. 652–662. AIAA 2005-4682.
- [17] Bowman, A. W., A. Azzalini, Applied Smoothing Techniques for Data Analysis, Oxford University Press, 1997.
- [18] B. Ganapathysubramanian, N. Zabarar, Sparse grid collocation schemes for stochastic natural convection problems, J. Comput. Phys. (2007), doi:10.1017/j.jcp.2006.12.014
- [19] W. Gautschi, Algorithm 726: ORTHOPOL- A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules, ACM Trans. Math. Software20, (1994) 2162.
- [20] E. Isaacson, H.B. Keller, Analysis of Numerical Methods, Dover, 1994.
- [21] R. Ghanem and A. Doostan, On the construction and analysis of stochastic models: characterization and propagation of the errors associated with limited data, J. of Comput. Phys. 217 (2006) 6381.
- [22] S. Fersin, R. B. Nelsen, et al. Dependence in probabilistic modeling, Dempster-Shafer theory, and probability bounds analysis, SAND Report 2004-3072.
- [23] W.L. Oberkampf and M. F. Barone, Measures of agreement between computation and experiment: validation metrics. Journal of Computational Physics, V. 217, 2006, pp. 5–36.

BIBLIOGRAPHY

- [24] Z. Sandor and P. Andras, Alternative Sampling Methods for Estimating Multivariate Normal Probabilities, Econometric Institute Report EI 2003,05
- [25] Iman, R. L. and Conover, W. J. (1980), Small Sample Sensitivity Analysis Techniques for Computer Models, with An Application to Risk Assessment, Communications in Statistics, A9(17), 1749-1842. Rejoinder to Comments, 1863-1874.
- [26] Wyss, G.D. and Jorgensen, K.H. (1998) A user's guide to LHS: Sandia's Latin hypercube sampling software. Available online at: <http://www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/1998/980210.pdf>
- [27] M. Loeve, Probability Theory, 4th ed., Springer-Verlag, New York, 1997.
- [28] F. Nobile, R. Tempone and C.G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. Report number: TRITANA 2007:07. To appear SINUM.
- [29] E. Novak, K. Ritter, Simple cubature formulas with high polynomial exactness, Constructive Approx., 15 (1999) 499-522.
- [30] Witteveen, J.A.S., Iaccarino, G., Simplex Elements Stochastic Collocation for Uncertainty Propagation in Robust Design Optimization 48th AIAA Aerospace Sciences Meeting, Orlando, Florida (2010) AIAA-2010-1313.

BIBLIOGRAPHY

- [31] Witteveen, J.A.S., Iaccarino, G., Simplex elements stochastic collocation in higher-dimensional probability spaces, 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, Florida (2010) AIAA-2010-2924.
- [32] Alonso, J. J., Hahn, S., Ham, F., Herrmann, M., Iaccarino, G., Kalitzin, G., LeGresley, P., Mattsson, K., Medic, G., Moin, P., Pitsch, H., Schluter, J., Svard, M., Van der Weide, E., You, D. and Wu, X., 2006, CHIMPS: A high-performance scalable module for multi physics simulations. AIAA Paper 2006 5274.
- [33] Drela, M., Xfoil: An Analysis and Design System for Low Reynolds Number Airfoils, Low Reynolds Number Aerodynamics (Conference Proceedings), edited by T.J. Mueller, University of Notre Dame 1989.
- [34] Tangler, J., Kocurek, J.D., Wind Turbine Post-Stall Airfoil Performance Characteristics Guidelines for Blade-Element Momentum Methods, NREL/CP-500-36900.
- [35] NWTC Design Codes (PreComp by Gunjit Bir). <http://wind.nrel.gov/designcodes/preprocessors/precomp/>. Last modified 26-March-2007; accessed 26-March-2007.
- [36] NWTC Design Codes (BModes by Gunjit Bir). <http://wind.nrel.gov/designcodes/preprocessors/bmodes/>. Last modified 20-March-2008; accessed 20-March-2008.

BIBLIOGRAPHY

- [37] NWTC Design Codes (FAST by Jason Jonkman, Ph.D.). <http://wind.nrel.gov/designcodes/simulators/fast/>. Last modified 05-November-2010; accessed 05-November-2010.
- [38] NWTC Design Codes (AeroDyn by Dr. David J. Laino). <http://wind.nrel.gov/designcodes/simulators/aerodyn/>. Last modified 31-March-2010; accessed 31-March-2010.
- [39] NWTC Design Codes (TurbSim by Neil Kelley, Bonnie Jonkman). <http://wind.nrel.gov/designcodes/preprocessors/turbsim/>. Last modified 25-September-2009; accessed 25-September-2009.
- [40] Moriarty, P., and Migliore, P., 2003 Semi-empirical aeroacoustic noise prediction code for wind turbines National Renewable Energy Laboratory
- [41] Brooks, T., Pope, D., and Marcolini, M., Airfoil Self-Noise and Prediction, NASA Reference Publication 1218, National Aeronautics and Space Administration, 1989.
- [42] Veldkamp, D., Chances in Wind Energy - A probabilistic Approach to Wind Turbine Fatigue Design. PhD thesis, Delft University.
- [43] Tuller, S.E., Brett, A.C., The characteristics of wind velocity that favor the fitting of a Weibull distribution in wind speed analysis. *Journal of Climate and Applied Meteorology*, 23:124134, 1984.

BIBLIOGRAPHY

- [44] Downey,R.P., Uncertainty in wind turbine life equivalent load due to variation of site conditions. Masters thesis, Technical University of Denmark, Fluid Mechanics Section, Lyngby, April 2006.
- [45] Lange,B., Modelling the Marine Boundary Layer for Offshore Wind Power Utilisation. PhD thesis, University of Oldenburg, December 2002.
- [46] Wieringa,J., Rijkoort,P.J., Windklimaat van Nederland (Wind Climate of the Netherlands). KNMI/Staatsuitgeverij, Den Haag, 1983.
- [47] Antoniou,I., Petersen,S.M., Højstrup,J. et al., Identification of variables for site calibration and power curve assessment in complex terrain. Technical Report JOR3-CT98-0257, Risø, CRES, WindTest, DEWI, ECN, Bonus and NEG Micon, July 2001.
- [48] Corten G. , Veldkamp H., Insects cause double stall, EWEC Copenhagen , 2001.
- [49] Corten, G.P., Insects Cause Double Stall, ECN-CX-00-018, Feb. 2001
- [50] Dyrmoose, S.Z., Hansen, P., The Double Stall Phenomenon and how to avoid it , IEA, Lyngby, 1998.
- [51] Madsen, H.A., Aerodynamics of a Horizontal Axis Wind Turbine in Natural Conditions, Risoe M 2903 1991.
- [52] Iachmann, H.S., Aspects of Insect Contamination in Relation to Laminar Flow Aircraft, Aeronautical Research Council current, April 1959.

BIBLIOGRAPHY

- [53] Croom, C. C.; and Holmes, B. J.: Flight Evaluation of an Insect Contamination Protection System for Laminar Flow Wings. SAE Paper 850860, April 1985.
- [54] Hardy, A. C.; and Milne, P. S.: Studies in the Distribution of Insects by Aerial Currents. *Journal of Animal Ecology*, vol. 7, 1938, pp. 199-229.
- [55] Crouch, J.D., Kosorygin, L.L. , Modeling the effects of steps on boundary layer transition, IUTAM Symposium on Laminar-Turbulent Transition, 2006.
- [56] Loeven, G.J.A. and Bijl, H., Airfoil Analysis with Uncertain Geometry using the Probabilistic Collocation method, Proc. of the AIAA 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Schaumburg (IL), United States, 2008.
- [57] Ilinca, F., Hay, A., and Pelletier, D., Shape Sensitivity Analysis of Unsteady Laminar Flow Past a Cylinder in Ground Proximity, Proceedings of the 36th AIAA Fluid Dynamics Conference and Exhibit , AIAA paper 2006 3880, San Francisco, June 2006.
- [58] Etienne, S., Hay, A., Garon, A., and Pelletier, D., Shape Sensitivity Analysis of Fluid-Structure Interaction Problems, Proceedings of the 36th AIAA Fluid Dynamics Conference and Exhibit , AIAA paper 2006 3217, San Francisco, June 2006.
- [59] Gumber, C. R., Newman, P. A., and Hou, G. J. W., Effect of Random Geometric Uncertainty on the Computational Design of a 3D Flexible

BIBLIOGRAPHY

- Wing, Proceedings of the 20th AIAA Applied Aerodynamics Conference, AIAA paper 2002 2806, St. Louis, June 2002.
- [60] Petrone, G., de Nicola, C., Quagliarella, D., Witteveen, J., Iaccarino, G., Wind Turbine Performance Analysis Under Uncertainty, AIAA. 49th AIAA Aerospace Sciences Meeting Including The New Horizons Forum And Aerospace Exposition. Orlando, 2011 (AIAA-2011-0544), 18 pp.
- [61] IEC 61400-1 (1999) Wind turbine generator systems-Part 1: Safety requirements, 2nd edition. International Electrotechnical Commission.
- [62] IEC 61400-1 (August 2005) Wind turbines-Part 1: Design requirements, 3rd edition. International Electrotechnical Commission.
- [63] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transaction on Evolutionary Computation, 6(2), pp. 181-197.
- [64] Deb, K., Sindhya, K., and Okabe, T., Self-adaptive simulated binary crossover for real-parameter optimization. In In Proceedings of the 9th annual conference on Genetic and evolutionary computation (GECCO '07). ACM, New York, NY, USA, 1187-1194., (2007).
- [65] Axerio-Cilies, J., Iaccarino, G., Petrone, G. and Sellapan, V., Extreme Ensemble Computation for Optimization Under Uncertainty , Proceedings of EUROGEN2011 Evolutionary and Deterministic Methods for Design, Optimization and Control, Capua, Italy, September 14-16, 2011 (ISBN 978-88-9063-230-3), pp 743-757.

BIBLIOGRAPHY

- [66] Waeschle, A., Cyr, S., Kuthada, T., and Wiedemann, J., 2004. Flow around an isolated wheel-experimental and numerical comparison of two cfd codes. Society of Automotive Engineers(2004-01-0445).
- [67] Mears, A., 2004. The aerodynamic characteristics of an exposed racing car wheel. PhD thesis, University of Durham, England.
- [68] Skea, A., Bullen, P., and Qiao, J., 1998. The use of cfd to predict the air flow around a rotating wheel. In 2nd MIRA international conference on vehicle aerodynamics, Birmingham, UK.
- [69] McManus, J., and Zhang, X., 2006. A computational study of the flow around an isolated wheel in contact with the ground. *Journal of Fluids Engineering*, 128, May, pp. 520–530.
- [70] Fackrell, J., 1974. The aerodynamics of an isolated wheel rotating in contact with the ground. PhD thesis, University of London.
- [71] Axerio-Cilies, J., Issakhanian, E., Jimenez, J., and Iaccarino, G. An aerodynamic investigation of an isolated stationary formula 1 wheel assembly. *Journal of Fluids Engineering*, 2011
- [72] Tang, Z., Periaux, J., Bugeba, G., Onate, E., Lift maximization with uncertainties for the optimization of high lift devices using Multi-Criterion Evolutionary Algorithms. *IEEE Congress on Evolutionary Computation 2009*: 2324-2331
- [73] Poloni, C., Padovan, L. Parussini, L., Pieri, S., Pediroda, V., Robust Design of Aircraft Components: a Multi-Objective Optimization Prob-

BIBLIOGRAPHY

- lem, von Karman Institute for Fluid Dynamics, Lecture Series 2004-07, November 15-19, 2004, ISBN 2-930389-56-7
- [74] Mattson, C. A. and Messac, A., Pareto Frontier Based Concept Selection under Uncertainty, with Visualization, Optimization and Engineering, Kluwer Publishers - Special Issue on Multidisciplinary Design Optimization, Invited Paper, Vol. 6, No. 1, 2005, pp. 85-115.
- [75] Petrone, G. Iaccarino, G., Quagliarella, D., Robustness Criteria in Optimization Under Uncertainty, Proceedings of EUROGEN2011 Evolutionary and Deterministic Methods for Design, Optimization and Control, Capua, Italy, September 14-16, 2011 (ISBN 978-88-9063-230-3), pp 586-603.
- [76] Rosenbrock, H. H. (1960), An automatic method for finding the greatest or least value of a function, *The Computer Journal* 3: 175-184, doi:10.1093/comjnl/3.3.175, ISSN 0010-4620, MR0136042
- [77] Schalk Kok, S., Sandrock, C., Locating and Characterizing the Stationary Points of the Extended Rosenbrock Function. *Evolutionary Computation* 17, 2009.
- [78] Goldstein, A. A. and Price, I. F.: On descent from local minima. *Math. Comput.*, Vol. 25, No. 115, 1971.
- [79] Eldred, M.S., Design under uncertainty employing stochastic expansion methods, *International Journal for Uncertainty*, Vol.1, No.2, 2011, pp. 119-146.

BIBLIOGRAPHY

- [80] Justesen, P., Ursem, R., Many-objective Distinct Candidates Optimization using Differential Evolution on centrifugal pump design problems. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010
- [81] Wang, J. F. and Periaux, J. (2001) Multi-Point optimization using GAS and Nash/Stackelberg games for high lift multi-airfoil design in aerodynamics, in Proceedings of the 2001 Congress on Evolutionary Computation CEC2001 (May 2001), pp. 552-559.
- [82] Petrone, G., de Nicola, C., Quagliarella, D., Witteveen, J., Axerio-Cilies, J., Iaccarino, G., Wind Turbine Optimization Under Uncertainty with High Performance Computing, 41st AIAA Fluid Dynamics Conference and Exhibit 27 - 30 Jun 2011. Honolulu, Hawaii (AIAA-2011-3806), 16 pp.
- [83] Petrone, G., de Nicola, C., Quagliarella, D., Witteveen, J., Iaccarino, G., Optimization of wind turbine performance and noise under uncertainty, Wind Turbine Noise 2011. Rome, 2011 (ISBN 978-88-88942-339) 16 pp.
- [84] Fonseca, C. M. and Fleming, J., An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1-16, 1995.
- [85] Zhong, B., Qiao, Z., Multi-objective optimization design of transonic airfoils, ICAS-94-2.1.1, 1994.

BIBLIOGRAPHY

- [86] Magoules, F., Fundamentals of grid computing: theory, algorithms and technologies, volume 11 of Chapman & Hall CRC numerical analysis and scientific computing. CRC Press, (2009).
- [87] Mahjoub, A., Pecero Sanchez, J., and Trystram, D., Scheduling with uncertainties on new computing platforms. Computational Optimization and Applications 48, 369398 (2011). 10.1007/s10589-009-9311-0.
- [88] D’Amato, E., Daniele, E., Mallozzi, L., Petrone, G., Tancredi, S., A hierarchical multi-modal hybrid Stackelberg-Nash GA for a leader with multiple followers game, in Dynamics of Information Systems: Mathematical Foundations, A.Sorokin and P. Pardalos Eds., Springer Proceedings in Mathematics, Springer, in press, 12 pp.
- [89] D’Amato, E., Daniele, E., Mallozzi, L., Petrone, N leader–M follower coalition games with genetic algorithms and applications, Proceedings of EUROGEN2011 Evolutionary and Deterministic Methods for Design, Optimization and Control, Capua, Italy, September 14-16, 2011 (ISBN 978-88-9063-230-3), pp 852-866
- [90] D’Amato, E., Daniele, E., Mallozzi, L., Petrone, Equilibrium strategies via GA to Stackelberg games under multiple follower’s best reply, International Journal of Intelligent Systems, (ISSN 1098-111X), in press, 14 pp
- [91] Nash, J., Non-cooperative games, *Annals of Mathematics* **54**, 286–295 (1951).

BIBLIOGRAPHY

- [92] von Stackelberg, H.F. (1934) *Marktform und Gleichgewicht* (Market Structure and Equilibrium), Julius Springer Vienna, 1934.
- [93] Denisova, L., Garnaev, A. (2008) Fish Wars: Cooperative and Non-Cooperative Approaches, *AUCO Czech Economic Review* 2 issue 1, 28-40.
- [94] Ben Abdelaziz, F., Ben Brahim, M., Zaccour, G. (2008) R&D equilibrium strategies with surfers, in *J. Optim. Theory Appl.*, 136 (2008), no. 1, 1-13.
- [95] Başar, T. and Srikant, R. (2002) Stackelberg network game with a large number of followers, Special issue in honor of Professor Jose B. Cruz, Jr. *J. Optim. Theory Appl.* 115 (2002), no. 3, 479-490
- [96] Marcotte, P. and Blain, M. (1991) A Stackelberg-Nash model for the design of deregulated transit system, *Dynamic Games in Economic Analysis*, Ed. by R.H. Hamalainen and H.K. Ethamo, *Lecture Notes in Control and Information Sciences*, Springer Verlag, Berlin, vol. 157, pp. 21-28.
- [97] Chinchuluun, A., Pardalos, P.M. and Huang, H-X. (2009) Multilevel (Hierarchical) Optimization: Complexity Issues, Optimality Conditions, Algorithms, in *Advances in Applied Mathematics and Global Optimization* (D. Gao and H. Sherali, Eds), Springer, pp. 197-221.
- [98] D'Amato, E., Daniele, E., Mallozzi, L., Petrone, G. and Tancredi, S. (2010) A hierarchical multi-modal hybrid Stackelberg-Nash GA for a leader with multiple followers game, Preprint n. 17 Dipartimento di Matematica e Applicazioni Universita di Napoli Federico II.

BIBLIOGRAPHY

- [99] Sheraly, H.D., Soyster, A.L. and Murphy, F.H. (1983) Stackelberg-Nash-Cournot equilibria: characterizations and computations, *Operation Research* vol. 31, pp. 253-276.
- [100] Jaynes, E. T., *Information Theory and Statistical Mechanics*, *Physical Review*, vol. 106, no. 4, pp. 620-630; May 15, 1957.
- [101] Shannon, C.E., Prediction and entropy of printed English, *The Bell System Technical Journal*, 30:50–64, January 1951.
- [102] Başar, T. and Olsder, G.J. (1999) *Dynamic noncooperative game theory*, Reprint of the second (1995) edition. *Classics in Applied Mathematics*, 23. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.
- [103] Luo, Z.-Q., Pang, J.-S., Ralph, D. (1996) *Mathematical programs with equilibrium constraints*, Cambridge University Press, Cambridge.
- [104] Mallozzi, L. and Morgan, J., On equilibria for oligopolistic markets with leadership and demand curve having possible gaps and discontinuities, *Journal of Optimization Theory and Applications* **125**, n.2, 393–407 (2005).
- [105] Rosen, J., Existence and uniqueness of equilibrium points for concave n-person games, *Econometrica* **33**, 520–534 (1965).
- [106] Gaudet, G. and Salant, S., Uniqueness of the cournot equilibrium: new results from old methods, *The Review of Economic Studies* **58**, 399–404 (1991).

BIBLIOGRAPHY

- [107] Liu, W. and Chawla, S. (2009) A game theoretical model for adversarial learning, {2009 IEEE International Conference on Data Mining Workshops, pp. 25-30.
- [108] Periaux, J., Chen, H., Mantel, B., Sefrioui, M., and Sui, H., Combining game theory and genetic algorithms with application to ddm-nozzle optimization problems., *Finite Elements in Analysis and Design* **37**, 417–429 (2001).
- [109] Vallée, T. and Başar, T. (2001) Off-Line computation of Stackelberg solutions with the genetic algorithm *Computational Economics*, vol. 13, pp. 201-209.
- [110] Breton, M., Alj, A., Haurie, A. (1988) Sequential Stackelberg equilibria in two-person games, *J. Optim. Theory Appl.*, 59 (1988), no. 1, 71-97.
- [111] Luo, Z.-Q., Pang, J.-S., Ralph, D. (1996) *Mathematical programs with equilibrium constraints*, Cambridge University Press, Cambridge.
- [112] Leitmann, G. (1978) On generalized Stackelberg strategies, *Journal of Optimization Theory and Applications*, Vol. 26 pp. 637-643.
- [113] Border, K. C. (1989) *Fixed point theorems with applications to economics and game theory*, Cambridge University Press.