

Massimiliano Vasile *Editor*

Optimization Under Uncertainty with Applications to Aerospace Engineering

 Springer

Optimization Under Uncertainty with Applications to Aerospace Engineering

Massimiliano Vasile
Editor

Optimization Under Uncertainty with Applications to Aerospace Engineering

 Springer

Editor

Massimiliano Vasile
University of Strathclyde
Glasgow, UK

ISBN 978-3-030-60165-2 ISBN 978-3-030-60166-9 (eBook)
<https://doi.org/10.1007/978-3-030-60166-9>

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

1	Introduction to Spectral Methods for Uncertainty Quantification	1
	João F. Reis, Giulio Gori, Pietro M. Congedo, and Olivier Le Maître	
2	Introduction to Imprecise Probabilities	35
	Daniel Krpelík and Tathagata Basu	
3	Uncertainty Quantification in Lasso-Type Regularization Problems	81
	Tathagata Basu, Jochen Einbeck, and Matthias C. M. Troffaes	
4	Reliability Theory	111
	Daniel Krpelík, Frank P. A. Coolen, and Louis J. M. Aslett	
5	An Introduction to Imprecise Markov Chains	141
	Thomas Krak	
6	Fundamentals of Filtering	181
	Cristian Greco and Massimiliano Vasile	
7	Introduction to Optimisation	223
	Annalisa Riccardi, Edmondo Minisci, Kerem Akartunali, Cristian Greco, Naomi Rutledge, Alexander Kershaw, and Aymen Hashim	
8	An Introduction to Many-Objective Evolutionary Optimization	269
	Dani Irawan and Boris Naujoks	
9	Multilevel Optimisation	307
	Margarita Antoniou and Peter Korošec	
10	Sequential Parameter Optimization for Mixed-Discrete Problems	333
	Lorenzo Gentile, Thomas Bartz-Beielstein, and Martin Zaefferer	
11	Parameter Control in Evolutionary Optimisation	357
	Margarita Antoniou, Rok Hribar, and Gregor Papa	

12	Response Surface Methodology	387
	Péter Zénó Korondi, Mariapia Marchi, and Carlo Poloni	
13	Risk Measures in the Context of Robust and Reliability Based Optimization	411
	Elisa Morales Tirado and Domenico Quagliarella	
14	Best Practices for Surrogate Based Uncertainty Quantification in Aerodynamics and Application to Robust Shape Optimization	429
	Christian Sabater	
15	In-flight Icing: Modeling, Prediction, and Uncertainty	455
	B. Arizmendi, M. Morelli, G. Parma, M. Zocca, G. Quaranta, and A. Guardone	
16	Uncertainty Treatment Applications: High-Enthalpy Flow Ground Testing	507
	Anabel del Val, Olivier Chazot, and Thierry Magin	
17	Introduction to Evidence-Based Robust Optimisation	541
	Gianluca Filippi and Massimiliano Vasile	

Chapter 1

Introduction to Spectral Methods for Uncertainty Quantification



João F. Reis, Giulio Gori, Pietro M. Congedo, and Olivier Le Maître

Abstract Spectral methods (SM) for uncertainty quantification are introduced. We start by introducing the transition between the deterministic and the stochastic frameworks, using the one-dimensional heat equation as an example. A simple Monte Carlo (MC) technique to solve the stochastic equation is introduced, together with its main advantages and drawbacks. The Karhunen–Loève expansion, a crucial tool to construct other (SM), is presented. Non-intrusive spectral projection (NISP) and Galerkin methods are introduced, and comparisons against the MC approach are discussed. The main differences between NISP and Galerkin methods are also highlighted. All the sections in the chapter are consistently illustrated with the one-dimensional heat diffusion problem.

Keywords Uncertainty quantification · Monte Carlo methods · KL expansion · Non-intrusive spectral method · Galerkin method

1.1 Motivation

One of the most challenging questions in science is to make predictions about a physical phenomenon of interest. We face this challenge in everyday life, since predictions are useful to plan our actions in advance and to optimise our choices to simplify our lives. Indeed, meteorological predictions help us deciding whether to organise a holiday trip in the countryside or not. Predictions regarding traffic at peak hours let us decide for different routes to avoid a long wait in a queue. In Wall Street, predictions drive brokers in choosing the most profitable investment.

J. F. Reis (✉) · G. Gori · P. M. Congedo
DeFI Team, CMAP Lab (École Polytechnique, Inria Saclay Île-de-France), Palaiseau, France
e-mail: joao.reis@inria.fr; giulio.gori@inria.fr; pietro.congedo@inria.fr

O. Le Maître
CMAP, CNRS, Inria, École Polytechnique, Palaiseau, France
e-mail: olm@limsi.fr

Engineers pledge their efforts to design devices to make our lives easier, and, to achieve this goal, they must predict the behaviour of the system they are creating.

Predicting the outcome of a process is a very challenging goal which in general implies the definition of a mathematical model. The equations included in the model describe the behaviour of the system under investigation (where the word system is entailed with its most general meaning), and their solution will yield to a prediction of the underlying outcome. To provide an example, this process is very much like translating a book from one language to another. Translation and modelling are so much alike that they share a common issue: information is lost. Indeed, it might be difficult to translate an English saying into Chinese. This is due to a different language structure and, even more subtle, to the fact that a saying is usually strongly related to the specific culture, which in principle is different from country to country. Therefore, information is often lost during the translation process.

In the same way, modelling a physical phenomenon is challenging, especially when its complexity grows. The mathematical equations lack information about the phenomenon they model, and, rather than an *exact*, they become an *approximate* description of physics.

The discrepancies found between model predictions and the actual phenomenon are referred to as the *model error*. The model error can be divided in two types: the *aleatory* and the *epistemic* error. The aleatory error is related to the randomness of physics under which the phenomenon develops. As exact physical conditions are impossible to measure, exact predictions of the outcome are also impossible to obtain. The epistemic error is instead related to a lack of knowledge regarding the real physics of the event, i.e., it is due to things one could in principle know but doesn't in practice. As a consequence, the equations included in the model may not be suitable to represent reality in a general sense. To make an example, when Newton wrote his famous equations for classical mechanics, he was not aware of the relativistic effects. His equations work perfectly in many cases, but they fail when the system under investigation consists of an object traveling at a velocity close to the light speed or when the object has a very large mass.

This is due to an epistemic uncertainty; the relativistic effects are not modelled; that affects Newton's dynamics, and it was not until Einstein fulfilled this deficiency that predictions about astronomical phenomena could be accurately made. Of course, the accuracy gained by modelling relativistic effects would not be worth the growth in complexity of the model itself in the limit of Newton's physics.

1.1.1 Typical UQ Questions

In real world applications, there exist many different questions that the methodologies presented in this chapter help to address.

Starting from the computation of statistical quantities, for instance, the mean and the variance of the output of a stochastic process, uncertainty quantification (UQ) techniques span from inference problems to data analysis. In this subsection, we

will try to summarise the different possible applications, and we will provide the reader with some thoughtful examples.

The most simple example consists in characterising a quantity of interest (QoI) within a given process. Roughly speaking, very often engineers are challenged with designing complex systems that are supposed to behave in a certain manner and under nominal conditions. One may think, for instance, of an aircraft that flies at a fixed altitude, with a well-defined cruise speed and with a given payload. The aircraft thus represents a system which depends on three inputs (air density, speed and payload), and a QoI may be the fuel consumption rate.

In real world applications, the atmosphere is not a homogeneous continuum. Turbulence, wind, clouds, and any meteorological phenomena contribute to modify the atmosphere through which the aircraft is flying, causing local fluctuations. Moreover, different payloads may be carried on board, possibly due to a different number of passengers or to a different mission scope.

Engineers are then concerned with the fact that the aircraft they are designing must face a wide range of different operating conditions. The operating conditions are not known a priori, if not as just parameter ranges or desired flight envelope. The goal of engineers is to come up with an aircraft which is able to safely accomplish the appointed missions, thus tolerating considerable variation of the operating conditions.

Therefore, engineers may want to understand how the uncertainties on the nominal flight conditions affect aircraft performances. They may, for instance, assess how a small variation in the cruising speed affects the efficiency and thus the fuel consumption. Uncertainty quantification techniques can indeed be used to propagate uncertainties through a computational fluid dynamics model and help characterising the QoI. Engineers may estimate the mean fuel consumption rate and its variance with respect of uncertainties on the value of speed, air density and payload (Passengers and cargo). This is known as *forward uncertainty propagation*.

Moreover, UQ techniques may be exploited to carry out sensitivity analysis. Sensitivity analysis is very useful when one is trying to assess the contribution of every single source of uncertainty to the variance of the QoI.

With reference to the aircraft example, a sensitivity analysis may be carried out to understand if the largest variations of fuel burning rate are related more to a variation in the payload than to the cruising speed. This information drives engineers during the design phase of the aircraft and allows them to achieve a more robust configuration.

Nonetheless, UQ techniques have a broad range of applications, and they can be exploited to compute the reliability of a device (or, more in general, a system). As known from the control theory, each device can be decomposed (up to a certain limit) into a collection of interacting sub-systems. A sub-system may be seen as an independent block that exchanges information through the connections from and towards other blocks (inputs and outputs). As such, each sub-system may undergo failures during its lifetime.

Due to the gradual deterioration of its components, the reliability of a device can be investigated as a time-dependent function, the *survival function*. This latter

object maps time to probability of functioning and provides an estimation of the device reliability in time.

Moreover, Bayesian techniques can be employed to infer quantities that cannot be directly measured. For instance, a broker working at the Wall Street stock market may be interested in pricing the assets of a company before deciding to buy/sell its shares, but there are no exact algorithms that allow to estimate the price of shares of a certain company. In order to achieve this task, one has to collect as much information as possible, and, most often, one has to come up with assumptions, as companies do not release information that are key to their business (such as industrial goals, market strategies, etc.).

Therefore, the price of shares cannot be measured, but it can only be inferred by gathering all the available information, including personal assumptions. Bayesian techniques, such as Bayesian networks [1], Bayesian inference methods and so on, may then be exploited in asset pricing problems, helping investors in the endeavour of increasing their capital.

Bayesian techniques also find a very interesting application in medicine. Indeed, very often doctors need to visualise the interior of a patient's body system in order to identify the disease and prescribe the most appropriate therapy. Of course doctors also aim at stressing the patient as little as possible, so they need to exploit the less intrusive techniques.

Computed tomography (CT), ultrasound, electrocardiogram (ECG) and Magnetic Resonance Imaging (MRI) are just a few examples of how information about the internal state of the human body is investigated. All these techniques rely on measurements, electric signals, magnetic-field variations, etc., which are not the QoI doctors are looking for. Through Bayesian inference techniques, it is possible to reconstruct images, even three-dimensional models, of an organ and thus provides information that helps identify the disease.

The examples reported in this section represent of course a small insight of a way broader world. Uncertainty quantification techniques may find their application in industrial processes, astronomy, physics, game theory, control theory, sociology and many others. In the rest of this chapter, we will try to refer to practical examples whenever it is possible.

1.2 Illustrative Problem

Throughout the chapter, we will make an extensive use of the heat diffusion problem to illustrate the advantages and the drawback of UQ techniques. This is intended to guide the reader in the journey of understanding the potential of UQ techniques.

The diffusion of heat through a continuum is of the utmost interest in many practical applications. The phenomenon has been deeply investigated since long time ago. A general study of the heat equation with applications can be found in [2, 3].

We start by introducing the deterministic model of the heat diffusion problem, providing a brief discussion about the boundary conditions and about the exact solution. Then, we evaluate potential sources of uncertainty, and we introduce the stochastic form of the heat diffusion equation. We close this section by pointing out some of the classical questions addressed by UQ techniques. We will answer to some of these questions along the rest of the chapter. Although this all process is focused on the simple illustrative problem, the same questions and procedures can be applied to more complex and general problems.

1.2.1 The Deterministic Heat Diffusion Equation

In the following section, we discuss the heat diffusion problem through a solid continuum. To keep it as simple as possible, we consider the propagation of heat across a straight beam, or, in other words, we consider a one-dimensional problem.

Assuming that we have a metal rod, our problem consists in modelling the distribution of the temperature $u(x)$. The one-dimensional heat diffusion problem reads

$$\rho(x) C_p(x) \partial_t u(x, t) - \partial_x k(x) \partial_x u(x, t) = \dot{q}(x, t), \quad (1.1)$$

where $x \in (0, 1)$ and $t \in (0, +\infty)$. In Eq.(1.1), ρ represents the density of the matter; C_p is the specific heat capacity, and k is the *thermal conductivity*. These are properties of the matter, and the three of them are always positive ($k, \rho, C_p \in \mathbb{R}^+$). These properties may be a function of space, and thus they vary along the beam, or they may be homogeneous. In general, their value can be inferred through experiments or predicted exploiting semi-empirical models. The \dot{q} term in Eq.(1.1) represents a distributed heat source (or sink) that models the production (or destruction) of heat within the domain (Fig. 1.1).

Given a thermodynamic system in a non-stable equilibrium state, the law of thermodynamics implies that if the system is perturbed by an infinitesimal disturbance, a process will necessarily occur. If the process is irreversible, as it is always the case in practical applications, after a certain amount of time, the system will reach a stable state.

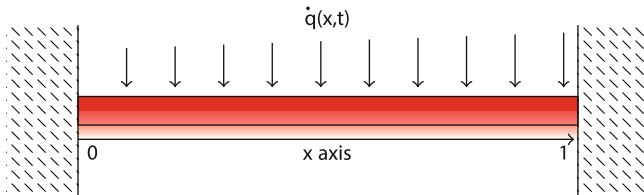


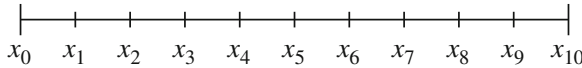
Fig. 1.1 Sketch of the heat diffusion along a one-dimensional beam

Hereafter, we will look for the steady solution of the heat diffusion problem only, which in fact corresponds to the stable equilibrium state that is necessarily reached after a sufficient amount of time.

This implies that we will consider the steady problem only, i.e. the system is non-transient, and the distribution of temperature does not depend on time. Moreover, we will assume that the source term is constant in time and homogeneous in space $\dot{q}(x, t) = f$. Under these hypotheses, the heat diffusion problem reduces to

$$\partial_x k(x) \partial_x u(x) = -f \quad x \in (0, 1) \quad (1.2)$$

Throughout this chapter, we will compute the solution of this differential equation, using a finite element method based on the Galerkin approach; see [4]. We therefore define the domain of length $l = 1$, and we divide it in N_{el}



For a given value of k , it is then possible to compute the deterministic solution to Eq. (1.2), provided that the boundary conditions are specified. Indeed, in order to fully specify the heat diffusion problem, we need to complement Eq. (1.2) with consistent boundary conditions, one for each of the beam edges. Boundary conditions may be of different types: Dirichlet (D) and Neumann (N). These boundary conditions have different physical meanings, and more details can be found in [2, 5]. To keep the problem as simple as possible, we will stick with Dirichlet boundary conditions, meaning that the temperature is prescribed at both ends of the beam. This yields the heat diffusion Dirichlet problem,

$$\begin{cases} \partial_x k(x) \partial_x u(x) = -f & x \in (0, 1) \\ u(0) = u_0 \\ u(1) = u_1 \end{cases} \quad (1.3)$$

For a homogeneous medium ($k(x) = \text{const}$), it is possible to retrieve the analytic solution of Eq. (1.2). Indeed, integrating both sides of the equation twice yields the following expression:

$$u(x) = -\frac{f x^2}{2k} + \left(u_1 + \frac{1}{2k} - u_0 \right) x + u_0. \quad (1.4)$$

When the thermal conductivity is not constant, Eq. (1.2) consists in a non-linear ordinary differential equation, and the analytic solution is not trivial.

1.2.2 The Stochastic Heat Diffusion Equation

The heat diffusion problem may be affected by a number of uncertainties. For instance, the temperature at one of the edges of the beam may be known within a given interval of confidence, or the length of the beam itself may be uncertain. To account for the uncertainty affecting a quantity, for instance, the temperature imposed at a boundary, one should avoid using a deterministic value and assign a probability density function to the variable.

In this section, we consider one source of uncertainty only: the one related to the thermal conductivity term k . Under the mentioned hypotheses, the stochastic heat diffusion model reads,

$$\begin{cases} \partial_x k(x, \theta) \partial_x u(x, \theta) = -f & x \in (0, 1), \quad \theta \in \Theta. \\ u(0, \theta) = u_0 \\ u(1, \theta) = u_1 \end{cases} \quad (1.5)$$

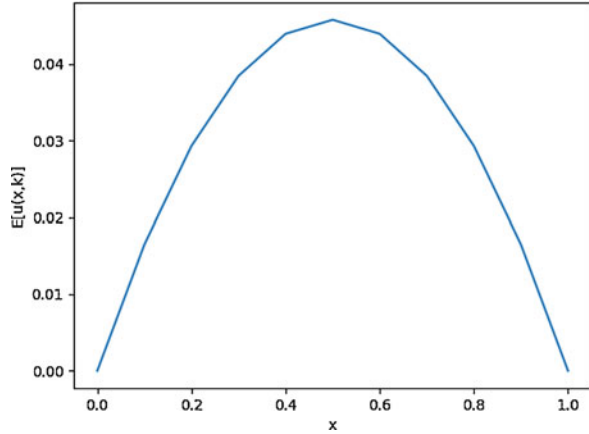
Equation (1.5) is a generalisation of Eq. (1.3), where $k(x, \theta)$ is a *random field* or a *stochastic field*. A random field is a map that receives two types of inputs: a random variable, in this case θ , and second variable from a deterministic space, in this case x . In other words, this means that $k(x, \theta)$ is a function that does not just depend in space, but it also depends on the random element θ . This random element θ could either be a variable or a vector, and it belongs to the set Θ . If θ is, say, a normal distributed variable, then $k(x, \theta)$ is a surface in $k(x, \theta) : (0, 1) \times \mathbb{R} \rightarrow \mathbb{R}$. Usually, the complexity of the physics implies that Θ is a multidimensional set. The dependency on θ is a choice we make a-priori, and the process to write $k(x)$ w.r.t. θ is called parametrisation. We will see later in this chapter how this parametrisation is done and how useful it will be.

In general, we are concerned with the evaluation of a QoI resulting from the solution of Eq. (1.5) rather than the solution itself. The QoI may be any statistical value, like the mean and the variance, or any higher statistical momentum. In some cases one could be interested in assessing the sensitivity of the solution with respect to an input parameter.

Some of this information can be easily computed when $k(x, \theta) \equiv k(\theta)$, i.e., k is homogeneous; thus it does not depend in space. Nevertheless, as a function of two variables, the fact that k is constant in x does not imply that it is a constant variable. Indeed, it is still a stochastic space that depends on the random parameter θ . For instance, using the solution in Eq. (1.4), we can readily compute the mean of higher moments of $u(x, \theta)$. For instance, if we consider a log-normal-distributed variable k with mean μ and variance σ , the mean solution (Fig. 1.2) is given by,

$$\mathbb{E}[u(x, k)] = -\frac{fx^2}{2\mathbb{E}[k]} + \left(u_1 + \frac{1}{2\mathbb{E}[k]} - u_0\right)x + u_0, \quad (1.6)$$

Fig. 1.2 Plot of the function in Eq. (1.6). This is the solution of Eq. (1.5) when $k \sim \text{lognormal}(0, 0.1)$. Space discretization has 11 points



where u_0 and u_1 are the left and right hand-side boundary conditions. As stated above, the solution plotted in Fig. 1.2 is obtained for the particular case $k(x, \theta) \equiv k \sim \text{lognormal}(0, 0.1)$. In particular, the equality $\mathbb{E}[1/k] = 1/\mathbb{E}[k]$ holds for such distribution. The problem we are referring to is very simple, and there exist well-established procedures to compute the statistical moments of its solution. On the other hand, if the problem was too complicated, it would be impossible to retrieve the very same statistical information through an analytic procedure. The only way to overcome the complexity of the problem would be to find an approximation of $\mathbb{E}[u(x, k)]$ using techniques, such as the Monte Carlo (MC) or spectral methods (SM).

1.3 The Sampling Process

The sampling process consists in the selection of one (or more) *individual*, from a well-defined set called a *population*. Each element is usually characterised by a certain number of properties; these properties are allowed to discriminate the population, according to specific criteria. For instance, we could be interested in characterising a country according to the age of its citizens or according to their educational level. We could be interested in identifying functioning/failed items in a single batch from an industrial production line, or we could classify flowers depending on the colour of their petals.

In descriptive statistics, the goal of the sampling process is to obtain a representative subset of individuals, the *sample set*, to estimate characteristics of the whole population. Indeed, one single sample will give us all the information about the random element we draw, but it will not tell us much about the whole population. If a sufficient number (in the statistical sense) of samples are collected, then the population can be characterised. The characterisation is strictly bound to

the considered population, and it depends on the property we are looking at, on the sampling procedure employed and on the size of the sample set.

So far we talked about characterising the population by looking at one property at a time. Nevertheless, quite often the properties of an individual are somehow related, like in case we were interested in characterising the population of school students according to their age and to their height. Not surprisingly, we will likely find out that people with a similar age will also have a similar height. Moreover, on average the older students are also the taller ones. This means that the parameters (age and height) are correlated, and the population is described by a joint probability distribution of these two variables.

According to the Pearson product-moment correlation coefficient, the definition of covariance for two random scalar variables takes the form

$$C(X, Y) = \frac{\mathbb{E}[X - \mathbb{E}[X], Y - \mathbb{E}[Y]]}{\sigma_X \sigma_Y} \quad (1.7)$$

where X and Y would be indeed the age and the height.

We now take advantage of the concept of stochastic field as an infinite and ordered collection of random variables. In practice, a random field may be alternatively seen as a variable that varies randomly and continuously over a N -dimensional domain. Velocity fluctuations in a turbulent flow, the height of microscopical ridges over rough surfaces, or the value of the thermal conductivity coefficient along a metal rod are examples of stochastic fields.

We consider a stochastic field $U(\mathbf{x}, \theta)$, where $\mathbf{x} \in \Omega$ is the spatial domain and $\theta \in \Theta$ is the probability (or stochastic) domain. For example, we can have the pair $\Omega := [0, 1]$ and Θ as the space of standard normally distributed variables. Moreover, we will refer to the response of the system as $U(\mathbf{x}, \theta)$.

Furthermore, we assume that all $\theta \in \Theta$ are somehow correlated; therefore, under the assumption that U is continuous in the mean square sense, we can define the correlation function C_{UU} as

$$C_{UU}(\mathbf{x}, \mathbf{x}') = \mathbb{E}[U(\mathbf{x}, \cdot), U(\mathbf{x}', \cdot)] \quad (1.8)$$

Therefore, the correlation function describes the statistical correlation between the values of the field at two different location in Ω . To this extend, the correlation function sometimes relies on the definition of a distance, the *correlation length* $l = |x - x'|$. The parameter l is defined in a particular domain (for instance, but not limited to, the temporal, the spatial or the frequency domains), and, in general, the correlation among two points decreases as their distance increases

$$\lim_{|x-x'| \rightarrow \infty} C_{UU}(\mathbf{x}, \mathbf{x}') = 0 \quad (1.9)$$

When the correlation function is relative to the same random variable at two different points, the term *autocorrelation function* is usually employed. Furthermore,

when the correlation function is relative to different random variables, it is usually referred to as *cross-correlation function*.

A classical example is a surface that is not perfectly flat, for instance, the Earth crust. We could look at this as a stochastic field where the height varies according to the terrestrial coordinates. To characterise the crust we could decide to sample the height of the ground, with respect to the sea level, at different locations.

We will refer to three exemplary cases: a plain flat region, a hill country and a rocky mountain. Once we map each region, we are left with three large data sets, and, from each of them, we are able to compute the mean ground altitude and its standard deviation.

If we look at the dataset related to the flat plain territory, we will find out that the altitude does not vary significantly in space, and the deviation of each sample is almost negligible (perhaps in the orders meters). This example is representative of a very large correlation length; if the altitude at one point is 100 m above the sea level, then the altitude of each point in the plain must be similar (otherwise it would not be a plain).

We leave the plain, and we travel to a different location, where the land is characterised by hills, and we repeat the very same procedure. We found out that the altitude of the ground can vary quite significantly (even a few hundred meters) on a very short distance (kilometers). So any measurement will give us an idea of what is the altitude of the portion of the ground around us. The knowledge about the altitude of the ground at one point tells us a lot about the altitude of the terrain in the close proximity of such point, but that will be a meaningless information if we are interested in the height of a point a few kilometers away. Correlation length is thus smaller than in the previous case, as the terrain is now more irregular, though the variation is still smooth.

We now move to the rocky mountains, and, once again, we repeat the measurements. In this case we will face edges, ridges, walls, cracks or cliffs. The ground altitude can therefore change abruptly, by hundreds of meters on a very short distance. In this rocky region, knowing the height of the crust at one point doesn't really tell us anything about the altitude of the ground around us. For instance, this is what happens when we are close to the edge of a cliff. The correlation length is then very small, and, to some approximation, we could assume that our measurements are uncorrelated.

Therefore, a very short correlation length implies that the information collected at one point (the altitude of a specific point over the Earth surface) doesn't give any information about the surrounding landscape. On the contrary, a very large correlation length allows us to have a wider perspective on the territory.

Note that the correlation length is not a property of the problem, but it is a parameter related to our knowledge. In this particular example, it represents the information that we have regarding the sample set, i.e., the type of territory from which the samples were taken. To make this point clear, one altitude sample would be sufficient to characterise an entire plain region, but, to be able to do that, we first need to know that the sample was taken in a flat territory. Therefore, if we were just

given the altitude value and not told that it had been sampled in a plain region, we could not have any idea of what the territory looks like.

It is needless to say that the very same reasoning applies to any stochastic field that we may find in any kind of application.

1.4 Sampling Techniques

The evolution of a physical process depends on parameters, the inputs, which are uncertain, and, in general, we are provided with their probability distributions in the stochastic space.

When we are dealing with a set of independent variables, we can just sample independently from each probability distribution, to reconstruct the vector of stochastic inputs. On the other hand, when the random variables are dependent, we should sample from their joint probability distribution. This is usually not straightforward, and we generally need to exploit special techniques.

For a jointly distributed set of variables $\theta \in \Theta$, where Θ is a d -dimensional set, it is possible to define a $(d \times d)$ correlation matrix where each entry corresponds to the correlation coefficient among the i -th and the j -th variable.

One way to accomplish the sampling of multiple variables could be to decompose the correlation matrix into the product of a lower triangular matrix L and its conjugate transpose, according to the Cholesky method. Once L is available, it is possible to retrieve the input random vector as

$$\theta = L\eta \tag{1.10}$$

where η is a $(d \times 1)$ -dimensional vector whose elements can be sampled independently from a normal distribution $\eta \sim \mathcal{N}(0, 1)$.

An alternative way to proceed with the sampling of jointly distributed variables is to rely on a parametrisation of the stochastic space. This means establishing a functional relation that maps a set of independent random variables, hereinafter referred to as the germ, to the initial mutually correlated parameters. Differently than the Cholesky decomposition approach, which leaves the dimension of the stochastic space unchanged, in some cases the parametrisation opens the path to an order reduction of the problem. This is particularly useful when dealing with stochastic fields.

1.4.1 Karhunen–Loève Expansion

The Karhunen–Loève decomposition (KL), or proper orthogonal decomposition (POD), was first proposed in the 1940s by Kac and Siegert [6], Karhunen [7] and Loeve [8]. The main idea upon which the KL expansion relies on is to provide

a representation of a stochastic model based on the spectral decomposition of its correlation function.

In general, it is possible to retrieve a linear operator K which is referred to as the *correlation kernel*. We will avoid diving into the mathematical details of this topic, the interested reader may refer to [9] to get a thorough and comprehensive presentation of the subject. What is relevant here is that K owns some very useful properties that allow us to retrieve a spectral expansion of the stochastic process U .

In particular, given its properties K has real non-negative eigenvalues λ_i . For each λ_i there exist a finite number of linearly independent eigenvectors $u_i(\mathbf{x})$. The collection of these eigenvectors constitutes the orthogonal basis upon which the KL expansion is built. Therefore, it is possible to decompose the kernel as follow:

$$K(\mathbf{x}, \mathbf{x}') = \sum_{i \geq 1} \lambda_i u_i(\mathbf{x}) u_i(\mathbf{x}') \quad (1.11)$$

This is a common eigenproblem that can be solved using well-established approaches.

Once the eigenvalues and their related eigenvectors are known, it is possible to retrieve the KL expansion of the stochastic problem U which reads

$$U(\mathbf{x}, \theta) = \sum_{i \geq 1} \sqrt{\lambda_i} u_i(\mathbf{x}) \eta_i(\theta) \quad (1.12)$$

It is worth noting that the random variables $\eta_i(\theta)$ that enter in Eq. (1.12) have zero mean, unit variance, and they are mutually uncorrelated.

This latter point is of the utmost importance. Indeed, if a certain stochastic process depends on a given set of mutually correlated variables, one could exploit the KL expansion to get a parametrisation of the stochastic space.

In particular, if the stochastic space includes a random field, the KL expansion contains an infinite number of terms. In practice, we truncate the expansion, trading some accuracy in place of a reduction of the dimensions of the stochastic space (and a reduction of the computational effort). The truncated series in Eq. (1.13) counts a number of terms at most equal to dimensions of the stochastic space, which in some case could be very large.

$$\hat{U}(\mathbf{x}, \theta) := \sum_{i \geq 1}^{N_{KL}} \sqrt{\lambda_i} u_i(\mathbf{x}) \eta_i(\theta) \quad (1.13)$$

In order to set the truncation order N_{KL} properly, one should make sure to include the contribution of all the significant eigenmodes associated with the correlation kernel. An important aspect that must be taken into account is that the rate of decay of the eigenvalues is strictly related to the correlation length. Figure 1.3 reports the eigenvalues, normalised w.r.t. the largest one, as resulting from the KL decomposition for different values of the correlation length l . The smaller

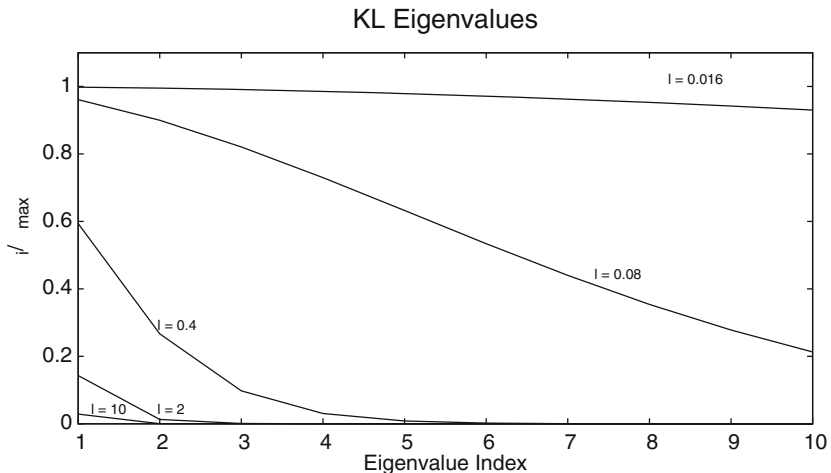


Fig. 1.3 Karhunen–Loève decomposition: eigenvalue decay rate, for different values of the field correlation length l

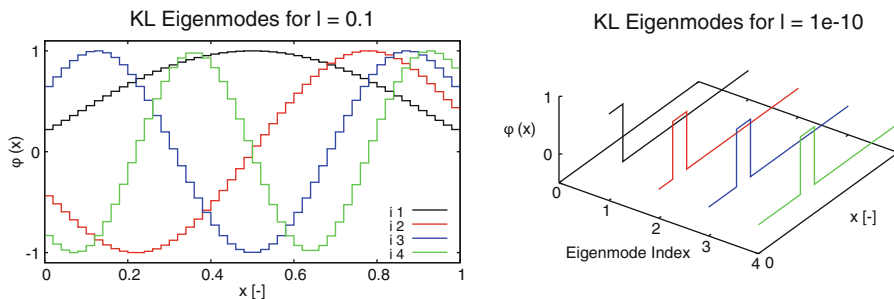


Fig. 1.4 Karhunen–Loève decomposition (a) eigenmodes for a covariance length $l = 0.1$ (b) eigenmodes for a very limited covariance length $l = 1e^{-10}$

the correlation length, the slower the eigenvalues decay. Therefore, in order to truncate the series and yet have a good level of approximation, we need to include a larger number of eigenmodes. On the other hand, a very large correlation length is associated with a fast eigenvalue decay which allows us to truncate the KL expansion at a lower order. To this extent, it is key to point out the fact that the KL expansion is optimal in the mean square sense, that is, the approximation of the process U , resulting from a truncated KL expansion, minimises the mean square error [9]. The reason for this is that the expansion is made onto a set of uncorrelated (hence, orthogonal) random variables. Moreover, not only the eigenvalue decay rate strictly depends on the correlation length, but also the shape of the eigenmodes is affected (Fig. 1.4).

Now that we have a parametrisation of the original stochastic space, we need to generate samples $\eta_i(\theta)$ in order to get $\hat{U}(\mathbf{x}, \theta)$. Here is where the assumption

that U is normally distributed is found to be valuable. Indeed, by linearity of Gaussian random variables, this assumption implies that $\eta_i(\theta)$ are jointly standard and normally distributed. Since they are uncorrelated, they are also independent, and to stress this we introduce the notation ξ_i for jointly normal independent random variables. Hereinafter, we will refer to $\xi := (\xi_1, \dots, \xi_{N_{KL}}) \in \mathcal{E}$ as the *germ*. Therefore, the probability space \mathcal{E} is the space of all jointly, normally and independent N_{KL} -dimensional random variables. In this perspective, the truncated KL expansion of a normally distributed stochastic process reads,

$$\hat{U}(\mathbf{x}, \xi) := \sum_{i \geq 1}^{N_{KL}} \sqrt{\lambda_i} u(\mathbf{x})_i \xi_i \approx U(\mathbf{x}, \xi). \quad (1.14)$$

With particular reference to the conductivity field $k(\mathbf{x}, \xi)$ included in our illustrative problem, the sampling process must implement the steps reported in Algorithm 1

Algorithm 1 Generate a sample set of the conductivity field $k(\mathbf{x}, \xi)$ using the KL expansion

- 1 Decompose the correlation function according to the KL approach;
 - 2 Establish the truncation order N_{KL} ;
 - 3 Generate N_{KL} independent samples ξ^j of the germ;
 - 4 Evaluate the truncated KL expansion at N_{el} spatial points x_j ;
-

1.4.2 Mathematical Reformulation of the Dirichlet Problem

The parametrisation of the stochastic field k in independent and identically distributed (IID) random variables allows us to reformulate the Dirichlet problem in Eq. (1.5). Indeed, this problem now depends on a set of independent random variables \mathcal{E} and reads,

$$\begin{cases} \partial_x k(x, \xi) \partial_x u(x, \xi) = -f & x \in (0, 1), \quad \xi \in \mathcal{E}. \\ u(0, \xi) = u_0 \\ u(1, \xi) = u_1 \end{cases} \quad (1.15)$$

As mentioned, the chapter is intended to provide an introduction of the fundamentals of UQ. To this extent, besides the mathematical aspects, we consider the understanding of the physics of the problem of the utmost importance. We will try to be loyal to our agreement with the reader, and we will try to keep the mathematical description at the simplest possible level. Nevertheless, a few essential functional spaces must be introduced in order to be rigorous.

Functional analysis is essential to UQ. A classic reference is [10], while applications of functional analysis on UQ can be found in [9, 11]

Considering the space of stochastic processes,

$$L^2(\mathcal{E}) := \{\varphi(\xi) : \int_{\mathcal{E}} \varphi(\xi)^2 d\mu(\xi) < \infty\}$$

where

$$d\mu(\xi) = f_{\mathcal{E}}(\xi) d\xi$$

and $f_{\mathcal{E}}(\xi)$ is the probability density function (PDF) of the multivariate standard Gaussian random variable ξ . Now, because of independence the PDF $f_{\mathcal{E}}(\xi)$ is the product of each PDF $f_{\mathcal{E}_i}(\xi_i)$ of each one-dimensional random variable ξ_i , i.e., $f_{\mathcal{E}}(\xi) = \prod_{i=1}^N f_{\mathcal{E}_i}(\xi_i)$. This would not happen, if the variable were not independent. It is important to understand that this makes all the theory much simpler and saves a great deal of computational effort. Indeed, it is one of many benefits of choosing a parametrisation, such as the KL expansion for normally distributed stochastic processes.

Remark 1.1 The notation $d\mu(\xi)$ represents the measure of the probability space. If the measure of \mathcal{E} is the one-dimensional standard Gaussian measure then,

$$\int_{\mathcal{E}} \varphi(\xi) d\mu(\xi) = \int_{-\infty}^{\infty} \varphi(\xi) \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} d\xi$$

Moreover, considering the space,

$$H_0^1(0, 1) := \left\{ \phi(x) : \int_0^1 \phi^2 dx < \infty \text{ and } \int_0^1 (\phi')^2 < \infty \text{ and } \phi(0) = \phi(1) = 0 \right\}$$

which is a Sobolev space. Spaces, such as $H_0^1(0, 1)$, are important, because a function $\phi \in H_0^1(0, 1)$ has peculiar properties. For instance, the Lax–Milgram theorem [5], which gives necessary and sufficient conditions for existence and uniqueness of solution for Eq. (1.5), is valid. These results are the backbone of UQ theory, and they should be kept in mind. In the following, superscript 1 implies that the first derivative of the function is squared and integrable, while the subscript 0 points out that the function vanishes at the boundary. The $(0, 1)$ refers to the domain in which the function is defined. Finally, we denote the $\{\phi_i\}_{i=1}^{\infty}$ an orthogonal basis of the space $H_0^1(0, 1)$.

The solution of Eq. (1.15) is a stochastic process $u(x, \xi) \in H_0^1(0, 1) \otimes L^2(\mathcal{E})$. This means that if we focus on the deterministic part of u , i.e., if we freeze the variable θ and let x free to vary, then $u(x, \cdot) \in H_0^1(0, 1)$. If we do the opposite and focus on the probabilistic part of u , then $u(\cdot, \xi) \in L^2(\mathcal{E})$.

In particular, if we consider the deterministic heat diffusion equation (1.1), the solution does not have any probabilistic component; therefore $u(x) \in H_0^1(0, 1)$.

1.5 Monte Carlo Methods

Monte Carlo (MC) methods are a broad class of algorithms that find their application in a wide variety of engineering fields. For instance, they are exploited to solve optimisation problems; they can be used to compute complex integrals, or they can be used to generate draws from a probability distribution.

The MC methods rely on a repeated random sampling to realise a large set of numerical experiments to obtain statistical information about the stochastic process. The main idea onto which a MC method is built consists in using randomness to solve a problem that might be deterministic in principle. In particular, the law of large numbers allows us to compute a QoI in the statistical sense. For instance, the expected value of a random variable, like the probability of getting head or tail when tossing a coin, may be estimated by simply running a large number of independent experiments (or realisations).

Given their simplicity, MC methods are often used as a brute-force approach to tackle problems that otherwise would be too difficult, or even impossible, to solve. Nowadays, there exists a broad family of MC algorithms that are used in science and engineering; we recall here the *importance sampling* often employed in statistical physics, the direct simulation Monte Carlo (DSMC) used in micro-fluidics problems or the Monte Carlo localisation (MCL) applied in autonomous robotics.

1.5.1 Mean and Variance

The goal of this section is to show how two fundamental QoIs, the mean and the variance of the model output, can be computed using an MC approach. We now focus on the heat diffusion equation (Eq. (1.15)) where the uncertainty is related to the thermal conductivity; k is parametrised using the KL method. Since a stochastic field is involved, an infinite number of parameters should be used in this parametrisation. As shown in Sect. 1.4.1, we are sometimes allowed to truncate the KL series to reduce the dimensions of the stochastic space. In particular, in the limit of a very large correlation length, we could parametrise the whole field with respect of a one single random variable, i.e. we could truncate the series at the first term. Otherwise, we would have faced a slower eigenvalue decay, and, as a result, we should have included a larger number, though still finite, of terms within the truncated KL expansion. Nevertheless, the procedure to compute the mean and the variance would essentially be the same.

The deterministic solution of the heat diffusion equation is found by assigning specific values of k_i at each x_i element employed in the Galerkin finite element approach. As reported in Algorithm 1, the KL expansion can be exploited to generate a set of N_{el} correlated k_i samples, starting from the germ. As the domain is represented through a finite discretisation, the dimension of the germ is at most

equal to the number of elements employed, but it can be reduced if the field is strongly correlated.

To apply the MC approach, the set of k_i is sampled M times, according to its KL parametrisation. For each m -th sample set, we deterministically compute the value of temperature $U^{(m)}$ at a specific location, say at the beam mid-point. $U^{(m)}$ is called the m -th *realisation* of the MC procedure. By the law of large numbers, the M realisations can be used to compute the statistical moments of the solution U . Obviously, from very few realisations, we cannot retrieve sufficient information to compute statistical meaningful quantities. As we increase their number, we extend the MC set and therefore improve the quantity of information available. In Fig. 1.5 we show that the mean and the variance converge to specific values as more information is gathered. In particular, it takes almost 20,000 realisations before the convergence is reached. Following this very simple procedure, even higher order statistical moments could be considered as well. Most importantly, we were able to retrieve statistical information without bothering about the complexity of the problem under investigation, although this one is really simple and relatively cheap to solve. Nevertheless, if our model consisted of a CFD simulation, the MC approach would still have been successful, but it would have been required to run 20,000 simulations, a task that could be too demanding to accomplish at a reasonable computational time. Of course, the same procedure may be carried out for any point along the beam, to obtain the mean temperature distribution and the variance at different locations.

Figure 1.5 shows how the predicted mean and variance converge to the very same values even if the MC set is different. Figure 1.6 reports the mean temperature (a) distribution and the related variance (b), with respect to the x axis. Moreover, according to the law of large numbers, after a certain number of MC realisations, the quantity of interest become independent from the sampling set. With reference to Fig. 1.6b, we can see that the variance follows a similar pattern: the maximum variability is found at the centre of the domain, while the BC enforces the solution at the edges so that the variability there is null.

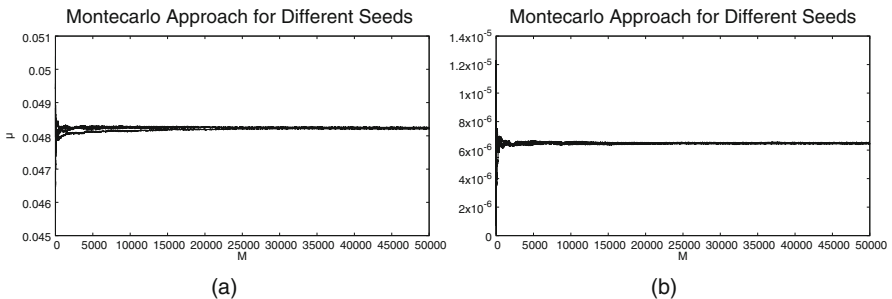


Fig. 1.5 MC approach: temperature mean (a) and variance (b) at $x = 0.5$, with respect of the number of realisations, for different initial seeds. The mean and variance of the stochastic field k are $\mu = 1$ and $\sigma^2 = 0.1$, respectively

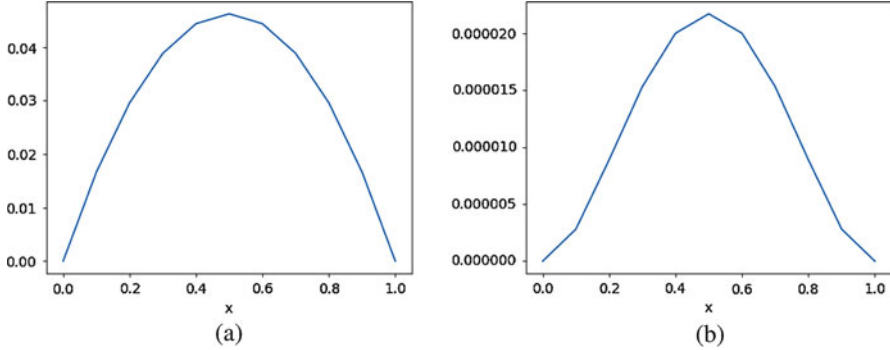


Fig. 1.6 Approximation of the QoI mean (a) and variance (b) using 10,000 MC samples. Same model and set up as in Fig. 1.2. The values of the mean and variance are, respectively, $\mu = 1$ and $\sigma^2 = 0.1$

Using the *Central Limit Theorem*, it can be proved that the convergence rate of MC methods for a well-behaved function corresponds to $1/\sqrt{M}$. It follows that, in order to halve the error, one has to multiply by four the number of realisations. This points out a very slow convergence rate that in some cases it may arise a few concerns about the strength of the approach.

1.5.2 PDF Reconstruction for Different Correlation Lengths

The MC approach may be employed also to compute the frequency of a QoI and to reconstruct its PDF. According to the KL expansion procedure, we were able to parametrise the thermal conductivity field, considering two opposite scenarios. The first one involves a field endowed with a very small correlation length $l \ll 1$, while in the second case $l \gg 1$. By applying the MC approach, we are able to draw an histogram with the relative frequencies of a QoI, the value of the temperature in the middle of the beam. The resulting histograms are reported in Fig. 1.7 where it is possible to point out how the frequencies are more or less spread over $[0, 1]$, depending on the magnitude of l . For both histograms in Fig. 1.7, we have considered k to have a log-normal covariance matrix, with mean $\mu = 1$ and variance $\sigma^2 = 0.1$. We clearly identify two different patterns for the two cases: a homogeneous conductivity field, i.e. a large correlation length, is characterised by a distribution of QoI spread over a wide support. On the other hand, a loosely correlated field is associated with a distribution with a smaller support. Most notably, the variance is much smaller than that related to a strongly correlated field.

Figure 1.7 also raises the following question: How does each random variable ξ_i influences the temperature at the middle point? Of course, there exist ways to address this question through a MC approach, but, given the usually slow convergence rate, this implies a great computational effort. In the next section,

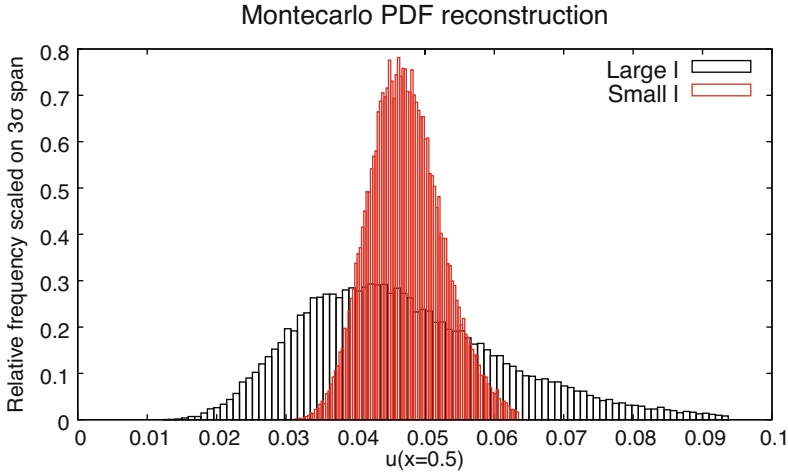


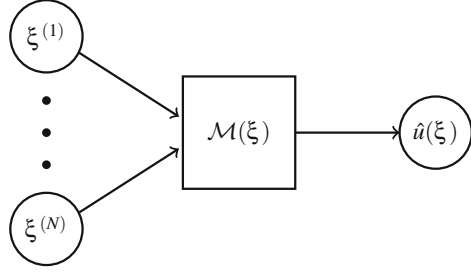
Fig. 1.7 PDF reconstruction for two different correlation cases: $l \ll 1$ and $l \gg 1$. We use $M = 10,000$ number of samples. The values of the mean and variance are $\mu = 1$ and $\sigma^2 = 0.1$, respectively

we introduce the Polynomial Chaos Expansions (PCE). These are polynomial expansions to approximate the solution of a stochastic model. The approach is somehow similar to what we showed for the KL expansion. In fact, both procedures are part of a class of methods called *Spectral Methods* (SM).

1.6 Spectral Methods

Monte Carlo approaches may conceal a very demanding effort, especially if the deterministic model is complex and if its evaluation requires a considerable amount of computational resources. Spectral methods represent an alternative approach to MC techniques. We introduce the concept of *surrogate* model or simply the *metamodel*. Once built, a surrogate represents an object capable of approximating the behaviour of the deterministic model with a satisfying accuracy and at a lower computational cost. Figure 1.8 represents the surrogate model as a black-box that receives a set of random inputs and returns the corresponding QoI, which is what the solution does itself. There exists different ways to construct a metamodel. In general, one has to choose a proper basis and to compute a set of coefficients that weight the chosen basis. For instance, the truncated KL expansion may be seen as a surrogate model, built over the spectral expansion of orthogonal (uncorrelated) functions of a stochastic field, being the weighting coefficient the eigenvalues. In the SM framework, the construction of a surrogate model relies on a polynomial expansion of the solution $u(\mathbf{x}, \xi)$.

Fig. 1.8 Sketch of the three stages toward computing a QoI \hat{u} . First generate a sample ξ , then build a surrogate model that depends on the germ ξ . Finally, we can use this surrogate to find \hat{u}



1.6.1 Polynomial Chaos Expansion

Polynomial Chaos Expansions (PCE) rely on the a priori assumption that the basis $\{\Psi_s\}$ in Eq. (1.17) is a polynomial of a given structure and order. This differs from the KL approach, where the basis consists of the eigenvectors related to the correlation kernel.

Recall the space of stochastic processes $L^2(\mathcal{E})$ introduced in Sect. 1.4.2. Denote the *generalised polynomial chaos basis* as the collection of the N -dimensional orthogonal polynomials $\{\Psi_s(\xi)\}_{s=0}^{\infty}$ up to order N_0 that benefits from the following property:

$$\mathbb{E}[\Psi_{s_1}(\xi)\Psi_{s_2}(\xi)] = \gamma_s \delta_{mn} \quad (1.16)$$

where $s_1, s_2 \in \{0, 1, \dots, P\}$ and P is the number of terms in the expansion. Finally, denote the normalising factor for each polynomial as $\gamma_s = \mathbb{E}[\Psi_s^2(\xi)]$.

The PCE is the functional dependency of the solution on the set of N IID random variables $\xi = (\xi_1, \dots, \xi_N)$ that reads:

$$u(\xi) = \sum_{s=0}^{\infty} u_s \Psi_s(\xi) \quad (1.17)$$

where the coefficients u_s are deterministic. The goal is then to find the coefficients u_s of such PCEs.

Equation (1.17) is a polynomial series, and, for practical applications, it should be truncated at a proper order N_0 . The advantage of the PCE is that the solution u can be accurately approximated using a relatively small number of terms.

The number of terms in the expansion, denoted by P , is related to N and N_0 by the following expression,

$$P + 1 = \frac{(N_0 + N)!}{N_0!N!}.$$

The truncated PCE of the stochastic process then reads

$$U(x, \xi) = \sum_{s=0}^P u_s(x) \Psi_s(\xi). \tag{1.18}$$

The N -dimensional polynomials Ψ_s are generated from the combination of one-dimensional polynomials ψ_s [9]. An example of the expression of a two-dimensional polynomial up to order 3, therefore $P_{nisp} = 9$, is presented in Table 1.1. The number of terms included in each polynomial increases with the order n . We are only interested in computing a realisation of Ψ_s . This means we do not compute the expression for $\Psi_s(\xi)$ but rather the realisation (real number) $\Psi(\xi_i)$. Of course the u_s and the $\Psi_s(\xi)$ exist also for higher orders and larger stochastic spaces. The integer s is related to a standard multi-index definition [9]. The N -dimensional polynomials Ψ_s are defined as a product of N one-dimensional ones. The order of this product is given by a multi-index related to s . In the previous example, since $N = 2$, each integer s corresponds to a two-dimensional vector (ξ_{n_1}, ξ_{n_2}) . This means Ψ_s is given by the product between one-dimensional polynomials of order n_1 evaluated at ξ_{n_1} and the polynomial of order n_2 evaluated at ξ_{n_2} . For instance, $s = 3$ corresponds to the vector $(0, 1)$. For practical purposes, we are not interested in the dummy variable definition of $\Psi_s(\xi)$ but rather in realisations of each polynomial for each order. With reference to Table 1.1, we see that the number of realisations for each polynomial increases with n . Finally, note that $\psi_0 = 1$, although it is explicitly reported in the polynomial expansion to highlight the recursive scheme.

There exists a one-to-one correspondence between the probability distribution of the germ and the type of polynomials one must chose as PCE basis. Table 1.2 reports the basic choice for different possible distributions of ξ . A more comprehensive dissertation on this topic may be found in [12].

Since the polynomial are fixed a priori, the goal is to compute the coefficients u_s of the expansion. It must be pointed out that SM become less efficient as the dimension N of the germ increases. This is the so-called *curse of dimensionality*. Indeed, the computation of the coefficients u_s may become too demanding, as

Table 1.1 The coefficients and the corresponding two-dimensional polynomials up to order 3. Each two-dimensional polynomial corresponds to the product of one-dimensional polynomials

ORDER	\mathbf{u}_s	$\Psi_s(\xi)$
$n = 0$	$[u_0]$	$[\Psi_0(\xi) := \psi_0 = 1]^T$
$n = 1$	$[u_{10} \ u_{01}]$	$[\psi_1(\xi_1)\psi_0 \ \psi_0\psi_1(\xi_2)]^T$
$n = 2$	$[u_{20} \ u_{11} \ u_{02}]$	$[\psi_2(\xi_1)\psi_0 \ \psi_1(\xi_1)\psi_1(\xi_2) \ \psi_0\psi_2(\xi_2)]^T$
$n = 3$	$[u_{30} \ u_{21} \ u_{12} \ u_{03}]$	$[\psi_3(\xi_1)\psi_0 \ \psi_2(\xi_1)\psi_1(\xi_2) \ \psi_1(\xi_1)\psi_2(\xi_2) \ \psi_0\psi_3(\xi_2)]^T$

Table 1.2 For each probability distribution characterising the stochastic variables, we are required to select a specific basis

Distribution	PCE polynomials	Support
Gaussian	Hermite	$(-\infty, \infty)$
Uniform	Legendre	$[a, b]$
Gamma	Laguerre	$[0, \text{inf})$
Beta	Jacobi	$[a, b]$

the number of realisations required to determine these deterministic coefficients increases exponentially, with the dimension of the germ. Moreover, the construction of N -dimensional polynomials gets more complex for higher order polynomials. This means that the computation complexity also increases with N_0 . Therefore, we need to find a balance between the accuracy and the computational cost.

There are several approaches to construct the PCE Eq. (1.17). In the following sections, we will present two different methods: Non-Intrusive Spectral Projection (NISP) and Galerkin methods.

As for the MC approach, NISP methods rely on deterministic evaluations of the considered model, to compute the PCE coefficients and to obtain the surrogate. Once the coefficients are available, the statistics of the output can be directly retrieved by established expressions. Indeed, there exist formulae that relate the coefficients to the statistics of the QoI. For instance, the coefficient u_0 is equal to the mean of the surrogate output.

1.6.2 Non-Intrusive Spectral Projection Methods

The goal of the chapter is to present clever techniques to retrieve certain information about a stochastic process. In general one could think of a model as a surjective mapping between the space of the parametrised input parameters and the QoI.

Non-intrusive spectral projection methods rely on the construction of a spectral expansion of the stochastic process. The surrogate approximates the behaviour of the original model in the sense that it is able to dictate a surjective mapping, up to a certain level of accuracy, between the stochastic input space and the output domain. The input parameters can be thus linked to the output through a functional relation.

The cost of employing NISP approach is associated with the amount of realisations needed to compute the deterministic coefficients in Eq. (1.18). For instance, coefficients may be computed using a quadrature formulae which reads,

$$u_n \approx \frac{1}{M} \sum_{j=1}^M w_j U(x, \xi^{(j)}) \Psi_n(\xi^{(j)}). \quad (1.19)$$

The number of required realisations equals the number of points $\xi^{(j)}$ in the quadrature formulae. These points can be chosen randomly or in a quasi-deterministic way, as discussion on the different quadrature approaches can be found later.

1.6.2.1 Numerical Approaches for NISP

The NISP algorithm is presented step-by-step in Algorithm 2. From the discussion above, the first and the fourth steps are intrinsically related. In the following we are going to discuss different sampling strategies that can be applied at step 1 of

Algorithm 2 NISP algorithm

- 1 Choose M samples of the germ $\xi^{(m)} \in \mathcal{E}$ with dimension N . The sampling strategy depends on step 4.
 - 2 Compute the M fields $K^{(m)}(x, \xi)$ using the truncated KL expansions in N terms.
 - 3 Compute M realisations $U^{(m)} = U(x, \xi^{(m)})$.
 - 4 Find the coefficients of the spectral expansion, i.e. solve Eq. (1.19).
-

Algorithm 2. The most straightforward way to evaluate the integral in Eq. (1.19) is through the MC approach. In order to do this, we impose $w_m = 1$ for all m and sample $\xi^{(m)}$, randomly. The strength of the MC quadrature approach is that it is not sensitive to the dimension of the stochastic space. This means that the computational cost will always be similar. The drawback is that the convergence is very slow, approximately equal to $1/\sqrt{M}$; see [13]. Therefore it requires a very large number of realisations to be unrolled. Nevertheless, there exist improved sampling strategies that one may employ to generate the M sample sets $\xi^{(m)}$ and that allows to increase the convergence rate.

Techniques, such as quasi-MC sampling and Latin-hypercube sampling (LHS) [9, 13], use the same unitary weights as MC methods, but the quadrature points are chosen in a deterministic way and not sampled randomly. However, these techniques are affected by the *curse of dimensionality*, as they depend on the dimension of the germ, and, as the stochastic dimension increases, their cost increases exponentially.

The LHS consists in forcing the sampler to draw the $\xi^{(m)}$ from equiprobable bins, in the parameter range. This yields to the construction of a Latin square grid, or a hypercube, if the stochastic space is multidimensional. In the hypercube there is one, and only one, sample along each axis-aligned hyperplane containing it. This LHS strategy allows to generate a near-random sample set of a multidimensional stochastic space.

Another way of computing the integral in Eq. (1.19) is using a Gauss quadrature formula where the samples $\xi^{(m)}$ and the weights w_m are selected deterministically. Generally, this yields a much faster convergence rate than a MC quadrature. However, M is strictly related to the dimension of the stochastic space, and, due to the curse of dimensionality, M grows exponentially as the dimension of \mathcal{E} increases. Of course, there exist ways to overcome this issue and to reduce the number of quadrature points. These techniques are called sparse Gauss quadrature formulae [9]. Another drawback of Gauss formulae is that the set of points and weights for a quadrature of degree N_q is not contained in the same set of a $N_q + 1$ -degree quadrature. This means that all the new sets of points must be computed, if a more accurate quadrature formula is needed. Again, there are techniques, for instance, adaptive Gauss quadrature, that helps avoiding this by recycling the quadrature points. See [9, 14] for a discussion on sparse and adaptive Gaussian quadrature.

1.6.2.2 Linear Regression

The *linear regression* approach is also widely used. The goal is to find the vector of coefficients $\mathbf{w} = (u_1, \dots, u_S)$ for Eq. (1.18). To do that, we first compute the minimisation sample points that solve the minimisation problem

$$\min_{\xi^{(j)}, j=1, \dots, M} \sum_{j=1}^M \left(r(\xi^{(j)}) \right)^2$$

where $r(\xi) := u(x, \xi) - U(\xi)$. In particular, the residual $r(\xi)$ is orthogonal to the space of solutions $L(\mathcal{E})$, as the number of samples M increases. In order to use a small number of samples, some algorithms for the selection of particular sets of minimisation points are available. The references [9, 14] include good first reviews on how to choose these minimisation sample set points.

Once the sample sets are chosen, we construct the matrix

$$\Psi = \begin{pmatrix} 1 & \psi_1(\xi^{(1)}) & \dots & \psi_P(\xi^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \psi_1(\xi^{(M)}) & \dots & \psi_P(\xi^{(M)}) \end{pmatrix}$$

and the *information matrix* $\Psi^T \Psi$. Then the coefficients \mathbf{w} are given by

$$\mathbf{w} = \left(\Psi^T \Psi \right)^{-1} \Psi^T \mathbf{U}$$

where $\mathbf{U} = (U^{(1)} \dots U^{(M)})$.

Usually, the regression method is robust for a number of evaluations of $2P \leq M \leq 3P$ (see [14]) using an appropriate choice to minimise the number of sample points. On the other hand, the size of the matrix Ψ is very large, at least twice as the number of PC terms. The matrix is also ill-conditioned, which in practice means that it is not recommended to be inverted directly.

1.6.3 Galerkin Methods

Similar to what was shown in the previous section, the goal of the Galerkin approach is to build a surrogate model of the form of Eq. (1.18). Once again, we need to find the coefficients entering the PCE. As we have seen, NISP methods rely on a quadrature rule to compute the spectral coefficients. To achieve the very same goal, the Galerkin approach requires the implementation of two different steps. First, the stochastic problem must be reformulated, to introduce the PCE of the solution into the model. The PCE is a spectral expansion of a random process. In a deterministic

setting, the terms in the spectral sum are a pair of eigenvalues and eigenvectors, where the latter depend only on the spatial coordinates. In the PCE case, there is a third term that depends on the probabilistic space. This term is a polynomial, and it is part of a basis of the probability space. Secondly, the coefficients are computed using a Galerkin projection: the residuals are computed and projected onto the orthogonal basis $\{\Psi_s\}$. This procedure yields to the construction of a linear system which must be solved to retrieve the deterministic coefficients of the PCE.

As mentioned, the stochastic Galerkin method is based on a reformulation of the problem. Therefore, the approach requires full access to the equations. It is worth to point out that this is not always granted, and, in some cases, it is even not feasible, for instance, when we are dealing with a CFD framework. In this latter case, the model is of the utmost complexity, since it is made up by algorithms, conditional jumps, specific implementation choices for specific problems and so on. Therefore, the possibility of exploiting a Galerkin approach for complex problems is often highly questionable.

In the following, we will focus on our case study, the heat diffusion problem, for which the Galerkin method is surely a suitable approach.

1.6.3.1 Weak Formulation and Deterministic Discretisation

Recall the mathematical formulation presented in Sect. 1.4.2. The goal of this subsection is to present the weak formulation and the corresponding finite element representation for both the stochastic and the deterministic problems. We rewrite the deterministic model in Eq. (1.3),

$$\begin{cases} \partial_x k(x) \partial_x u(x) = -f & x \in (0, 1) \\ u(0) = u(1) = 0. \end{cases} \quad (1.20)$$

The weak formulation [2, 5] reads,

$$a(u, v) = b(v) \quad \text{for any } v \in H_1^0(0, 1) \quad (1.21)$$

where

$$a(u, v) = \int_0^1 k(x) \nabla u(x) \nabla v(x) dx \quad \text{and} \quad b(v) = \int_0^1 f(x) \nabla v(x) dx$$

The weak form is given by

$$\mathcal{A}(u, v) = \mathcal{B}(v), \quad \text{for any } v \in H_1^0(0, 1) \otimes L^2(\Theta) \quad (1.22)$$

where

$$\mathcal{A}(u, v) = \mathbb{E}[a(u, v)] \quad \text{and} \quad \mathcal{B}(v) = \mathbb{E}[b(v)]$$

We focus on the LHS of weak forms of Eqs. (1.21) and (1.22). The objects $a(u, v)$ and $\mathcal{A}(u, v)$ are called *functionals*. The functional $a(u, v)$ depends only on the space $H_1^0(0, 1)$, since for this case u and v are deterministic functions. On the other hand, the functional $\mathcal{A}(u, v)$ depends on the product of spaces $H_1^0(0, 1) \otimes L^2(\Theta)$, as now u and v are stochastic processes.

We proceed with a finite element discretisation to obtain a system of equations. This is the only discretisation we need to perform for the deterministic case of Eq. (1.21). However, for Eq. (1.22), we still need to perform a stochastic discretisation, as we will see later. The finite element discretisation is done over N_{el} points in $(0, 1)$. Indeed, the weak form of Eq. (1.21) yields the system

$$\mathbf{A}\mathbf{u} = \mathbf{B} \tag{1.23}$$

where the entries of \mathbf{A} and \mathbf{B} are given by

$$A_{i,j} := \int_0^1 k(x) \nabla \phi_i(x) \nabla \phi_j(x) dx, \quad \text{and} \quad B_j = \int_0^1 f(x) \nabla \phi_j(x) dx$$

Similarly, the stochastic form yields the system

$$A(\theta)\mathbf{u} = \mathbf{B} \tag{1.24}$$

where the entries of $A(\theta)$ differ from Eq. (1.23). We arrived to the main point of this description: Eq. (1.23) is a linear system of equations. On the contrary, Eq. (1.24) is not, since $A(\theta)$ is a stochastic matrix. Therefore, we need to discretise the matrix $A(\theta)$. To do that we follow the approach reported in [9] that brings us to the construction of a finite dimensional matrix \mathbf{A} that approximates $A(\theta)$.

1.6.3.2 Stochastic Discretisation

Denote $\mathbf{u} = [\mathbf{u}_0 \mathbf{u}_1 \dots \mathbf{u}_P]$ a block-vector where $\mathbf{u}_s \in \mathbb{R}^{N_{el}}$. The goal is to build a linear system that returns the coefficients \mathbf{u}_s of the spectral expansion

$$U(x, \xi) \approx \sum_{s=0}^{P_G} \mathbf{u}_s \Psi_s(\xi) \tag{1.25}$$

Recall the orthogonal basis of $L^2(\Theta)$ given by $\{\Psi_s\}$. First, find a spectral expansion of the field $k(x, \theta)$,

$$k(x, \theta) \approx \sum_{s=0}^{P_G} k_s(x) \Psi_s(\theta) \quad (1.26)$$

Moreover, define the third tensor $C \in \mathbb{R}^{(P_G+1)^3}$ by

$$C_{i,j,s} := \int_{\Theta} \Psi_i \Psi_j \Psi_s d\mu(\theta). \quad (1.27)$$

Finally, the block matrices $\mathbf{A}_{i,j}$ are given by

$$\mathbf{A}_{i,j} = \sum_{s=0}^{P_G} M^s C_{i,j,s} \quad (1.28)$$

where $M_{i,j}^s := \int_0^1 k_s(x) \nabla \phi_i(x) \nabla \phi_j(x) dx$ are matrices of size $N_{el} \times N_{el}$. It is important that the expansion of Eq. (1.26) has the same terms as each block matrix in Eq. (1.28); otherwise the products are not defined. This yields a linear system of size $N_{el}(P_G + 1) \times N_{el}(P_G + 1)$ given by

$$\mathbf{A}\mathbf{u} = \mathbf{B}. \quad (1.29)$$

where the $N_{el} \times N_{el}$ dimensional vector $\mathbf{B}_j = B$ for $j = 0, \dots, P_G$. The system can be presented as

$$\begin{bmatrix} \mathbf{A}_{0,0} & \mathbf{A}_{0,0} & \dots & \mathbf{A}_{0,0} \\ \mathbf{A}_{1,0} & \mathbf{A}_{1,1} & \dots & \mathbf{A}_{1,P_G} \\ \vdots & \vdots & \vdots & \ddots \\ \mathbf{A}_{P_G,0} & \mathbf{A}_{P_G,1} & \dots & \mathbf{A}_{P_G,P_G} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{P_G} \end{bmatrix} = \begin{bmatrix} B \\ B \\ \vdots \\ B \end{bmatrix}$$

The cost to solve the linear system in Eq. (1.29) is dramatically larger than the cost required to solve the deterministic one in Eq. (1.23).

1.6.3.3 Computational Cost of Stochastic Galerkin Method

The Galerkin method is the most computationally demanding method presented in this chapter. A good background on linear algebra and matrix analysis techniques is a must for reducing the cost of implementing this method. A simple brute-force approach is usually not an option, even for one-dimensional simple problems, such as Eq. (1.5). Nevertheless, this is a good starting point for more refined techniques. In Algorithm 3, we present such brute-force strategy to build the linear system of Eq. (1.29).

Algorithm 3 Stochastic Galerkin algorithm to compute the PCE coefficients

- 1 Compute the coefficients $k_s(x)$ of the expansion in Eq. (1.26).
 - 2 Compute matrices M_s for each $k_s(x)$.
 - 3 Compute the tensor C in Eq. (1.27).
 - 4 Compute the block matrices in Eq. (1.28).
 - 5 Solve the linear system of Eq. (1.29).
-

One way of computing the coefficients $k_s(x)$ is by using the KL expansion of $k(x, \xi)$. This means we need to write the expansion in Eq. (1.13) in the form of Eq. (1.26), and then we can use these coefficients to build the matrices M_s . However, there are ways of computing these matrices directly from the KL expansion of $k(x, \xi)$; see [9]. Next, we need to compute the third order tensor C that has $(P_G + 1)^3$ entries. We can do so by exploiting the orthogonality properties of the polynomials $\Psi_s(x)$. This yields a symmetric and sparse tensor that makes the procedure much more efficient. Finally, after completing step 4 we solve the linear system in Eq. (1.29). This is a sparse system of size $N_{el}(P_G + 1) \times N_{el}(P_G + 1)$. Again, there are a number of techniques one could exploit to solve this system in a more efficient way. These include the use of Krylov-based methods and pre-conditioning techniques [9].

1.6.4 Application of Surrogate Models: A Sensitivity Analysis Using PC Expansions

We can exploit the coefficients of the PCE of a model to get statistical information about the QoI, and this is a great improvement in terms of efficiency, if we were to use a MC approach, instead. In this section, we discuss the computation of the Sobol indices, a set of parameter which is very useful for sensitivity analysis. We do not provide a detailed explanation about how to compute Sobol indices, the interested reader can refer to [15] for details on this. The goal is to present a general overview about differences in computing the Sobol indices using a MC approach or a surrogate model approach. We also compute the Sobol indices for a specific example, using the surrogate model approach, as in [9, 16].

Let $\xi := \{\xi_1, \dots, \xi_N\}$ be a sample of the germ, and consider the solution value at the midpoint of the beam for this particular sample, i.e. $u^* = u(x = 1/2, \xi)$. We are interested in the significance of each random variable ξ_i *alone* with respect to the value u^* . This information is provided by the first order Sobol indices.

Let σ^2 be the variance of u^* . This variance accounts for the variability of all random variables ξ_i that contributed to the PCE of u^* . The variance σ^2 can be decomposed into parameters that stress the contribution of each variable to the QoI, in this case u^* . For instance, consider the stochastic space illustrated in Table 1.1. Here, we consider a two-dimensional stochastic space, and we decompose the variance of the solution as follows:

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \sigma_{1,2}^2, \quad (1.30)$$

where σ_1^2 and σ_2^2 are the contribution to the variance of ξ_1 and ξ_2 , respectively, which are independent random variables. The term $\sigma_{1,2}^2$ represents a second order contribution that accounts for combined interaction of the variables of the germ. In this subsection, we are interested in the first order Sobol indices only, i.e. the values of σ_1^2 and σ_2^2 .

1.6.4.1 MC Approach

If we ought to compute the values σ_1^2 and σ_2^2 using a MC approach, we would need to sample M values of ξ_1 (being ξ_2 fixed) and compute the corresponding M realisations. Then we could compute the variance of u^* for each ξ_1 , using the corresponding samples. This gives σ_1^2 using the corresponding estimators [15]. A similar procedure should be done for the second stochastic variable. An example with $M = 5$ samples illustrated below.

1. Generate M samples of 4-dimensional points ($2d$, $d = 2$) in the unit hypercube and construct the matrix M :

$$M = \begin{pmatrix} 0.3 & 0.3 & 0.7 & 0.3 \\ 0.2 & 0.3 & 0.8 & 0.4 \\ 0.2 & 0.1 & 0.5 & 0.4 \end{pmatrix}$$

2. Define the matrices A and B in the following way. The columns of A are the first d columns of M . The columns of B are the remaining columns of M .

$$A = \begin{pmatrix} 0.3 & 0.3 \\ 0.2 & 0.3 \\ 0.2 & 0.1 \end{pmatrix} \quad B = \begin{pmatrix} 0.7 & 0.3 \\ 0.8 & 0.4 \\ 0.5 & 0.4 \end{pmatrix}$$

3. Construct matrices $A_B^{(i)}$, $i = 1, \dots, M$ in the following way. The columns of matrix $A_B^{(i)}$ are the columns of matrix A except column i , which is the i -th column of B .

$$A_B^{(1)} = \begin{pmatrix} 0.7 & 0.3 \\ 0.8 & 0.3 \\ 0.5 & 0.1 \end{pmatrix} \quad A_B^{(2)} = \begin{pmatrix} 0.7 & 0.3 \\ 0.8 & 0.3 \\ 0.5 & 0.1 \end{pmatrix}$$

4. Compute the M -dimensional vector $U^{(A)}$, $U^{(B)}$, $U^{(i)}$, where each entry is the solution of Eq. (1.15) using the points (ξ_1, ξ_2) of the corresponding matrix. For

instance,

$$U^{(1)} = \begin{bmatrix} u(x, \xi_1^{(1)} = 0.7, \xi_2^{(1)} = 0.3) \\ u(x, \xi_1^{(2)} = 0.8, \xi_2^{(2)} = 0.3) \\ u(x, \xi_1^{(3)} = 0.5, \xi_2^{(3)} = 0.1) \end{bmatrix}$$

5. Use the estimators [15] to compute the Sobol indices. For instance, the first order Sobol indices are given by

$$S_i = \frac{\mathbb{V}_i}{\mathbb{V}} \quad (1.31)$$

where \mathbb{V} is some approximation of σ^2 and

$$\mathbb{V}_i = \frac{1}{N} \sum_{j=1}^M U_j^{(B)} \left(U_j^{(i)} - U_j^{(A)} \right).$$

It is clear that this turns out to be a quite computationally demanding algorithm when the number of samples M is high. Indeed, we need to find $M(d + 2)$ realisations to compute Eq. (1.31). We saw in Sect. 1.5 that the number of samples needed is of the order of thousands. In the next subsection, we present a different approach to compute the Sobol indices using the PCE as a surrogate.

1.6.4.2 Surrogate Approach

There are many ways of computing the Sobol indices using a surrogate model. In the following, we shall consider a PCE obtained using the NISP method. We notice that the coefficients are obtained using a quadrature rule. We also highlighted that the number of realisations of the solution is related to the number of quadrature points used. For a sufficient accurate Gauss–Hermite quadrature rule, the number of points should be much less than the number of realisations M previously used in the MC approach; therefore, the number of realisations for NISP should be much less than M .

In this subsection, we give an explicit formula to compute the Sobol indices. We also illustrate the meaning of these coefficients with a numerical example. Consider the multi-index $\mathbf{m}^j = \alpha_j \cdot \xi$, where the multi-index $\alpha_j \in \{0, 1\}^N$ is given by

$$\begin{aligned} \alpha_1 &= (1, 0, 0, \dots, 0) & \alpha_{Q_1+1} &= (1, 1, 0, \dots, 0) \cdots \alpha_{Q_N} = (1, \dots, 1, 1, 1) \\ \alpha_2 &= (0, 1, 0, \dots, 0) & \alpha_{Q_1+2} &= (1, 0, 1, \dots, 0) \cdots \\ & \vdots & & \vdots \\ \alpha_{Q_1} &= (0, 0, 0, \dots, 1) & \alpha_{Q_2} &= (1, 0, 0, \dots, 1) \cdots \end{aligned}$$

where $Q_q = \sum_{i=1}^q \binom{N}{i}$. As an example, consider $N = 3$ then $\mathbf{m}^3 = (0, 0, 1) \cdot (\xi_1, \xi_2, \xi_3) = (0, 0, \xi_3)$, which means $\sigma_{\mathbf{m}^3}^2 := \sigma_3^2$ which is the contribution of the variable ξ_3 to the overall variance σ^2 . Finally, we just need to define the set of indices $s \in \{1, \dots, P_{nisp}\}$ that are related to the variable(s) in m^j , $j = 1, \dots, Q_N$. Before formally defining this multi-index, we look at the case of Table 1.1. Since $N = 2$, we have 3 multi-indices \mathbf{m}^j , $j = 1, 2, 3$. One corresponds to ξ_1 , the other to ξ_2 and the last to both of them. By construction, the polynomials $\Psi_s(\xi)$ with index $s = 1, 3, 6$ depend only on ξ_1 ; the ones with index $s = 2, 5, 10$ depend only on ξ_2 ; and, finally, the polynomials with indices $s = 4, 7, 8$ depend both on ξ_1 and ξ_2 . Therefore, we can define the following three sets, according to the different dependences in the germ:

$$S_{m^1} := \{1, 3, 6\}$$

$$S_{m^2} := \{2, 5, 9\}$$

$$S_{m^3} := \{4, 7, 8\}$$

The general definition of S_{m^j} is given by

$$S_{m^j} := \left\{ s \in \{1, \dots, P_{nisp}\} : \Psi_s = \prod_{i=1}^N \psi_{m^i}(\xi_i) \right\}$$

Now, we can write the decomposition of the variance, using this notation:

$$\sigma^2 = \sum_{j=1}^{Q_N} \sigma_{\mathbf{m}^j}^2 \quad (1.32)$$

where each $\sigma_{\mathbf{m}^j}^2$ is given w.r.t. the coefficients of the PCE in Eq. (1.18) as

$$\sigma_{\mathbf{m}^j}^2 := \sum_{s \in S_{m^j}} u_s^2. \quad (1.33)$$

From Eq. (1.33) one can use the estimators as in [9, 16] to compute the Sobol indices and even other quantitative indices. In Table 1.3, we can see the first order Sobol indices of each variable ξ_i for different correlation lengths l . We have the same number of stochastic dimension as spatial elements, and therefore, we can clearly appreciate their importance on the solution. For smaller correlation lengths, all the stochastic variables have a significant importance on the QoI. This means that the KL expansion should include all possible terms ξ_i ; otherwise, the surrogate will not be accurate. On the other hand, if the correlation length is large, then the only significant variable is the ξ_i . This is because the only significant mode in the KL expansion is exactly the first one.

Table 1.3 First order Sobol indices for each random variable of the germ w.r.t. the value of the solution of Eq. (1.5) at the middle of the beam, $u^* = u(x = 1/2, \xi)$. The space discretisation with 11 elements. Numerical values for a value of the mean $\mu = 1$ and variance $\sigma^2 = 0.1$. The maximum order of the orthogonal basis is $N_0 = 2$, which gives an expansion with 18,591 terms

l	0.016	0.08	0.4	2	10
S_1^1	$3.835e - 01$	$4.761e - 01$	$9.535e - 01$	$9.996e - 01$	$9.999903e - 01$
S_2^1	$7.572e - 03$	$1.033e - 02$	$6.187e - 03$	$2.224e - 04$	$8.755946e - 06$
S_3^1	$4.610e - 01$	$4.483e - 01$	$3.829e - 02$	$5.235e - 05$	$8.145184e - 08$
S_4^1	$4.838e - 03$	$5.332e - 03$	$7.525e - 06$	$5.605e - 13$	$5.386858e - 20$
S_5^1	$6.338e - 02$	$3.673e - 02$	$1.332e - 06$	$4.395e - 11$	$1.173323e - 16$
S_6^1	$3.1801e - 03$	$1.403e - 03$	$2.411e - 10$	$5.364e - 23$	$1.067134e - 19$
S_7^1	$4.503e - 02$	$1.257e - 02$	$8.945e - 09$	$4.863e - 17$	$6.450506e - 20$
S_8^1	$3.712e - 03$	$3.772e - 04$	$6.991e - 16$	$3.020e - 21$	$3.854629e - 21$
S_9^1	$8.364e - 03$	$5.787e - 04$	$3.555e - 12$	$2.801e - 21$	$2.506504e - 26$
S_{10}^1	$3.540e - 03$	$5.780e - 05$	$9.602e - 23$	$2.181e - 26$	$3.183518e - 26$
S_{11}^1	$3.641e - 03$	$1.241e - 04$	$4.027e - 16$	$2.445e - 26$	$2.839943e - 26$

1.7 Concluding Remarks

Physical phenomena are intrinsically affected by uncertainties. Therefore, the mathematical models should also account for these uncertainties. In the chapter, we provide an elementary example of such physical phenomena—heat diffusion through a beam—and the corresponding model, Eq. (1.5). We illustrate a few UQ questions with this example. More than the solution of the model, we are interested in its QoI. We show two different ways of doing this: *à la* MC method and using SM. Spectral methods are appropriate methods to perform these computations, given their relatively cheap cost. In fact, complex models, such as the ones arising from fluid dynamics, have many sources of uncertainties, and MC methods are generally too demanding.

Heat diffusion through a beam may have different sources of uncertainty. In the chapter, we only consider a random conductivity field, but other uncertainties may be considered. We chose to parametrise the conductivity field, but other sampling approaches could have been pursued. The main reason why we chose to follow this path is because, given the assumptions on the random field, the KL expansion provides a parametrisation in independent random variables. This is important, because this implies that the surrogate given by the PCE is also depending on independent variables.

Therefore, SM are usually used, since their exponential rate of convergence compensates the complexity. However, if the parametrisation of the sources of uncertainty uses too many random variables, SM can perform poorly, given the larger complexity of the problem. In this case, MC methods may be better suited, given that its convergence does not depend on the number of independent variables in the parametrisation.

One of the main goals of this chapter was to demonstrate the versatility of a surrogate to find QoI. We demonstrated how this surrogate can be obtained by SM. Providing that we can parametrise the conductivity field with a small enough number of random variables, NISP or Galerkin methods are efficient ways to obtain a PCE of the solution of a model, such as Eq.(1.5). Once making the effort of computing this surrogate, the QoI is obtained almost “for free” using the coefficients of the PCE. We illustrate this by computing the Sobol indices w.r.t. to each random variable of the germ.

References

1. D.S. Sivia, *Data Analysis: A Bayesian Tutorial*, 1st edn. (Oxford University Press, Oxford, 2006)
2. S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*. UNITEXT - La Matematica per il 3+2 (Springer, Cham, 2012)
3. F. Saleri, A. Quarteroni, R. Sacco, *Numerical Mathematics*. Texts in Applied Mathematics, 1st edn. (Springer, New York, 2000)
4. A. Valli, A. Quarteroni, *Numerical Analysis of Partial Differential Equations*. Springer Series in Computational Mathematics, 1st edn. (Springer, Berlin 2008)
5. L.C. Evans, *Partial Differential Equations* (American Mathematical Society, Providence, 2010)
6. M. Kac, A.J.F. Siebert, An explicit representation of a stationary Gaussian process. *Ann. Math. Stat.* **18**(3), 438–442 (1947)
7. K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Number ARRAY(0x4deaf30) in Suomalaisen Tiedeakatemia toimituksia. Suomalainen Tiedeakatemia, Helsinki, 1947
8. M. Loeve, Fonctions alatoires du second ordre, in *Processus Stochastique et Mouvement Brownien*, ed. by P. Lévy (Gauthier Villars, Paris, 1948)
9. O.P. Le Matre, O.M. Knio, *Spectral Methods for Uncertainty Quantification: With Applications to Computational Fluid Dynamics*. Scientific Computation (Springer, Dordrecht, 2010)
10. R.G. Bartle, *The Elements of Integration and Lebesgue Measure* (Wiley, New York, 1995)
11. T. Sullivan, *Introduction to Uncertainty Quantification*, vol. 63 (Springer, Berlin, 2015)
12. D. Xiu, *Numerical Methods for Stochastic Computations: A Spectral Method Approach* (Princeton University Press, Princeton, 2010)
13. D. Gamerman, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. Texts in Statistical Science, 1st edn. (Chapman & Hall/CRC, Boca Raton, 1997)
14. G. Blatman, Adaptive sparse polynomial chaos expansions for uncertainty propagation and sensitivity analysis. PhD thesis, Université Blaise Pascal BLAISE - Clermont II, Institut Français de Mécanique Avancée et Université Blaise Pascal, Oct 2009
15. I.M. Sobol, Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **55**(1–3), 271–280 (2001)

16. K. Tang, P.M. Congedo, R. Abgrall, Adaptive surrogate modeling by anova and sparse polynomial dimensional decomposition for global sensitivity analysis in fluid simulation. *J. Comput. Phys.* **314**, 557–589 (2016)

Chapter 2

Introduction to Imprecise Probabilities



Daniel Krpelík and Tathagata Basu

Abstract Since uncertainty is persistent in engineering analyses, this chapter aimed to introduce methods to describe and reason with under uncertainty in various scenarios. Probability theory is the most widely used methodology for uncertainty quantification for a long time and has proven to be a powerful tool for this task. Nevertheless, the construction of stochastic models relies on very fine information, such as large amount of observations, which is not always available. Without it, the constructed models are only very rough approximations of the real laws and may cause incorrect decisions. In this chapter, we introduce other types of models, based on the theory of imprecise probability, which we are able to construct and reason with under situations with limited available knowledge.

Keywords Imprecise probability · Uncertainty · Lower previsions · Robust inference

2.1 Introduction

The desired outcome of an engineering project is a system which provides the service it was designed for. But the exact future behaviour of a system in the real world is, ipso facto, unknown, until the system is built and tested. This also applies to the use of familiar systems that operate under novel environmental conditions. However, this poses a dilemma: how to design systems so that they meet our requirements once deployed? Thus, we need some procedure(s) to help us with the

D. Krpelík (✉)

Department of Mathematical Sciences, Durham University, Durham, UK

Department of Applied Mathematics, VŠB - Technical University of Ostrava, Ostrava, Czechia

e-mail: daniel.krpelik@durham.ac.uk

T. Basu

Department of Mathematical Sciences, Durham University, Durham, UK

e-mail: tathagata.basu@durham.ac.uk

design process. Here, we identify two major aspects that should be included in such a procedure.

The first aspect is the ability to assess consequences of various actions—making predictions of the future behaviour. Science, per se, is a field that explores relations between various aspects of reality and constructs models upon which we may base our predictions. But there is no guarantee that these models are totally accurate. Mathematical models are usually simplifications of the occurring phenomena, additional simplifications often various need to be employed to make the computation tractable, and the numerical evaluation itself may introduce additional error (e.g. when simulating processes described by differential equations). Furthermore, another type of error is introduced when providing numerical inputs for the models, i.e. their parameters. These also come from scientific inference and, therefore, suffer from similar issues to those of the models themselves. Their usual sources are measurements with finite resolution, statistical inference from finite dataset and expert elicitation. All these are subjected to uncertainty. Therefore, regardless of the chosen model, its predictions are always subject to uncertainty, and this fact needs to be considered.

The second aspect is how to choose a single design from a set of admissible possibilities. The usual way to tackle this problem is to describe what it means when one solution is preferred over the other and then search for a solution which is preferred over all alternatives. Imagine for a while (and please drop this assumption later) that we can predict the consequences without a doubt. What would then constitute an “optimal” design? Most importantly, we would like the system to provide the service it has been designed for. Designs which ensure this are preferred over those which do not (see Chap. 4). But such a definition of preference generally fails to identify a single design because there are usually many ways to ensure the desired service. One could then introduce other desirable properties; e.g. a system is preferred over another if it is cheaper to realise, or when it is more environmentally friendly, or when it is easier to maintain, or when it produces greater volume of outputs in less time, or due to some other criteria. With such a definition, one can formulate, mathematically, a constrained optimisation problem and use standard algorithms to find its solution. But the preference criteria may be contradictory (e.g. cost vs performance), so the optimisation problem could generally not have a unique solution, and the procedure would yield a set of incomparable designs. This happens, for example, during a multi-objective optimisation. The solution to the multi-objective optimisation problem is a so-called *Pareto set*, which consists of solutions which are *better* than those excluded, but none is strictly preferred over the others in the set (see Chap. 8). Besides, once we drop the assumption that we can make perfect predictions, the optimum yield based on the erroneous model might not be the true optimal design we were searching for.

The main focus of this chapter lies in the first mentioned aspect—how to model the uncertainty associated with our assessments. Nowadays, this field is dominated by two complementary theories, probability theory and interval arithmetic. Although these two allow us to model many scenarios, they suffer from

several drawbacks, which make their application questionable in practical scenarios. Interval arithmetic is often overly conservative and fails to capture correlations among quantities of interest. Probability theory requires us to specify how likely the occurrence of each possible outcome is, which can be impossible up to the required level of precision needed to construct the mathematical models.

To overcome the issues with these formalisms, we will demonstrate the theory, which results from their flourishing marriage. We will introduce *imprecise probability* (IP) theory.

The history of imprecision in probabilistic assessments dates back to Boole's work on inductive logic [8, Chap. 18]. Imprecise probabilities could also be identified in several attempts to obtain bounds on probabilistic assessments when precise values were intractable (e.g. Markov's, Jensen's and Chebyshev's inequalities). Some early examples may also be found in the field of sensitivity analysis for statistical inference [6]. Nevertheless, by the mid-twentieth century, a separated theory of imprecise probabilities began to emerge as a generalisation of probability theory. This would, not exclusively, include the introduction of non-additive measures by Choquet [10], generalisation of statistical inference by Dempster [16], Walley's work on statistical inference with imprecise probabilities [40], and development of the theory of lower previsions [39].

In this chapter, we intend to show the basic ideas and structures behind IP theory together with some examples of its application.

2.2 Some Models of Uncertainty

We intend to begin the chapter with a practical example to show how uncertainty may be modelled and how it influences the quality of our predictions. This will be demonstrated on a simple, analytically solvable decision problem. We will show solutions given by various models and highlight their differences, but also their similarities.

For the rest of this section, we will be interested in the following scenario.

Example 2.1 Suppose that there exists an area, which is polluted. Such pollution will slowly deteriorate over time. Our question is, what is the earliest time when we can send people there without risking their health?

Let this be the set of information available to us without *a doubt*:

- Pollution level is known at time $t = 0$, say $u(0) = 1$.
- The highest pollution level, which does not pose any danger to human health, $u_s \in \mathbb{R}$, is also known.
- The pollution level decreases according to a known relationship,

(continued)

Example 2.1 (continued)

$$d_t u(t) = -au(t), \quad (2.1)$$

where a denotes a model parameter.

We will further explore how the predictions on the pollution level and the decisions about the time for sending people to the area change by varying the quality of knowledge about the model parameter a .

2.2.1 A Point Estimate

A common scientific practice is to identify unknown quantities by *point estimates*. These represent our *best guesses*. Model parameters may sometimes be known without a doubt. In other cases, such form might come from statistical procedures. If the parameter a from Example 2.1 is regarded as known exactly (is identified as a single real number), the prediction about the pollutant concentration at any non-negative time t is given by the unique solution of equation (2.1),

$$u(t; a) = \exp(-at), \quad (2.2)$$

which is, again, a single, precise value, $u(t; a) \in \mathbb{R}$, for each time $t \geq 0$.

Thus, what would we do if we were to make a decision with this predictive model? To guarantee personnel safety, the pollution level must be below the critical level u_s . We are looking for the smallest t , which is the earliest time, for which these criteria are met. This question can be translated into a mathematical optimisation problem,

$$\min_{t \geq 0} t \quad \text{s.t.} \quad u(t) \leq u_s. \quad (2.3)$$

Our model gives us the answer:

- “The smallest safe time t for visiting the area is $-\frac{\ln(u_s)}{a}$ ”.

2.2.2 An Interval

Point estimates may be overly optimistic in many cases. For example, in manufacturing processes, the geometry of the final device can be specified only up to known tolerances and allowed deviations. Similarly, due to discretisation of scales on our measuring devices, even direct measurement actually provides only bounds

for possible values of observed quantities. It is also often easier for experts to specify some “credible bounds” for a parameter, instead of a precise value. In all these cases, and many others, the uncertain variables of interest are only known to belong to some set, with no further preferences of credibility among its elements. Let us therefore return to the previous example and investigate what would happen if we were to know only that the model parameter a from the Example 2.1 lies in a set Ω_A .

As will be the case also in the next types of uncertain parameter specifications, the qualitative nature of the parameter will be carried, *propagated*, through the model and provide an answer of similar quality. If we propagate an *imprecise* parameter through a deterministic model, the model will, generally, give us imprecise answers. In Eq. (2.1), it will be a set of *credible* pollutant levels. Our least informative assessment about $u(t; a)$ is the image of the union of its arguments’ domains, $u(t, \Omega_A) := \{u(t, a) | a \in \Omega_A\}$. For real-valued functions, considering that these sets are intervals, and describing the uncertainty by the lower and upper bounds on the quantity of interest (QoI) often suffice. Translated to $u(t)$ from Example 2.1, our assessment about $u(t)$ will take the form

$$u(t) \in \left[\inf_{a \in \Omega_A} u(t; a), \sup_{a \in \Omega_A} u(t; a) \right] \stackrel{\Delta}{=} [\underline{u}(t), \bar{u}(t)]. \quad (2.4)$$

Let us assume that we know that $a \in \Omega_A = [\underline{a}, \bar{a}]$. For the process in Example 2.1, we will exploit that the function $u(t; a)$ is monotone (decreasing) in a for all t . Thus, the extremes will be attained on the set boundary. Given the explicit solution (Eq. (2.2)), we may therefore judge that

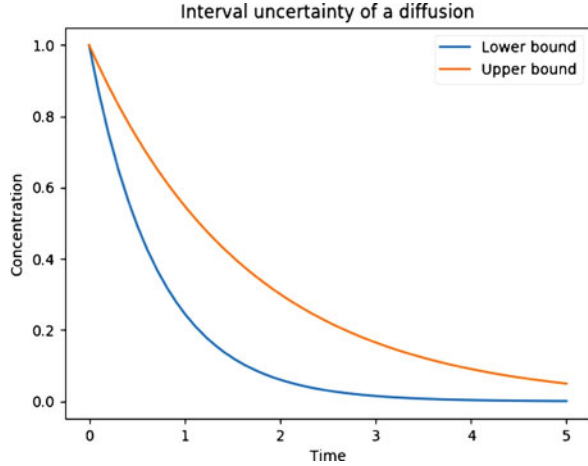
$$\begin{aligned} \forall t : \quad u(t) &\in [u(t; \max\{a \in \Omega_A\}), u(t; \min\{a \in \Omega_A\})] \\ &= [\exp(-\bar{a}t), \exp(-\underline{a}t)]. \end{aligned} \quad (2.5)$$

Time evolution of the pollution is presented in Fig. 2.1 via the lower and upper bounds.

A remark: Through imprecision, we are actually able to model a wider set of problems without introducing any additional computational complexity. Consider a dynamical process described by ordinary differential equation $d_t \tilde{u}(t) = -a(t)\tilde{u}(t)$ with initial condition $\tilde{u}(0) = 1$. The explicit solution is $\tilde{u}(t) = \exp\left(-\int_0^t a(x)dx\right)$. Let us further assume that the exponential rate is bounded at each time, i.e. $\forall t : a(t) \in [\underline{a}, \bar{a}]$. Because

$$\tilde{u}(t) = \exp\left(-\int_0^t a(x)dx\right) < \exp\left(-\int_0^t \underline{a}dx\right) = \exp(-\underline{a}t) = \bar{u}(t),$$

Fig. 2.1 Bounds for $u(t)$ when a is only known to belong to an interval



the bounds for $\tilde{u}(t)$ are the same as that for the simpler model. Similar enveloping properties of interval arithmetic are exploited in IP theory (see, e.g. Chap. 5).

Reasoning with imprecisions introduces some additional challenges. In the case of precise models, our logic provides clear answers to comparative (e.g. $x > y$) and inclusive (e.g., Is $x \in \Omega$?) statements, such as the following: they are either true or false. If we know only that a variable belongs to a set, we may arrive to indeterminate statements.

Consider that a model predicts that some QoI $x \in [1, 2]$. We can still determinate precise values of statements, such as $x > 0$ or $x > 5$, but we would be indecisive about, e.g. statement $x > 1.5$, which is possible, but not certain.

There is no general way to validate these statements for all the cases. Many computer algorithms require us to provide a means of comparing any two values. An example might be an optimisation algorithm, which needs to compare multiple solution proposals (although the problem might be treated as a multi-objective optimisation; see Chap. 8). A possible solution is to define an artificial ordering by comparing them by their upper ($x > y \Leftrightarrow \bar{x} > \bar{y}$) or lower ($x > y \Leftrightarrow \underline{x} > \underline{y}$) bounds, which is called the Γ -maximax and Γ -maximin criteria, respectively. By the transitivity of total orderings, both these methods include the determinate case $\underline{x} > \bar{y} \Rightarrow x > y$ but treat the indeterminate case differently.

Consider that we, again, want to determine the earliest safe time to enter the contaminated area from Example 2.1. We now assume interval uncertainty about the input parameter a , so the model predictions also result in intervals, by Eq. (2.5). We can employ both Γ -maximax or Γ -maximin orderings (because precise values may also be seen as degenerate intervals) and, depending on our choice, we arrive to either of the following:

- “The smallest safe time t for visiting the area is $-\frac{\ln(u_s)}{a}$ ” in the pessimistic (Γ -maximin) case, which actually guarantees compliance with the regulations.

- “The smallest safe time t for visiting the area is $-\frac{\ln(u_s)}{a}$ ” in the optimistic (Γ -maximax) case, which provides the earliest time for which compliance with regulations is *possible*, but not assured.

2.2.3 A Probability Distribution

Probability theory is a dominant framework to address uncertainty in science and engineering. When an outcome of an experiment cannot be determined uniquely from the available information, probability theory aims to formulate a *law* which models the behaviour of repeated outcomes from *identical* trials.

Apart from modelling of the repetition of trials, probability theory also provides a consistent reasoning framework, an extension to boolean logic [26]. With probability theory, we may encode our *degree of faith* in logical statements as probabilities (e.g. $x \in [1, 2]$ as $Pr(x \in [1, 2])$) and utilise probability theory to also obtain consistent *degrees of faith* for derived statements (e.g. $Pr(f(X) < 3)$). There are various ways to construct these models, ranging from statistical inference to elicitation by domain experts.

If a model parameter is a random variable (RV), the model predictions themselves are treated as RVs too. Especially when neither the random parameter nor the function are bounded, without further assumptions, we cannot construct any reasonable bounded set of *credible* values of the argument to carry out the best–worst case scenario inferences as in Sect. 2.2.2 (however, a heuristic construction is possible via confidence intervals and credible sets). Without loss of generality, we can assess the distribution of our predictions and back our further decisions on probabilistic logic.

Let us consider Example 2.1 again. Now, we will assume that the parameter a is a RV, A (denoted by a capital letter), distributed according to the exponential law $P_A(\cdot)$ with rate parameter λ and cumulative distribution function (CDF)

$$F_A(x) := Pr(A < x) = 1 - \exp(-\lambda x).$$

We can straightforwardly express the distribution of $U(t)$ (again denoted by a capital letter to emphasise that it is a RV) as

$$Pr(U(t) \in E) = P_A(\{a : u(t; a) \in E\}).$$

This will be further formalised in Eq. (2.7). Due to monotonicity, we get a CDF for every $U(t)$, the $F_{U(t)}$, as (Theorem 2.1)

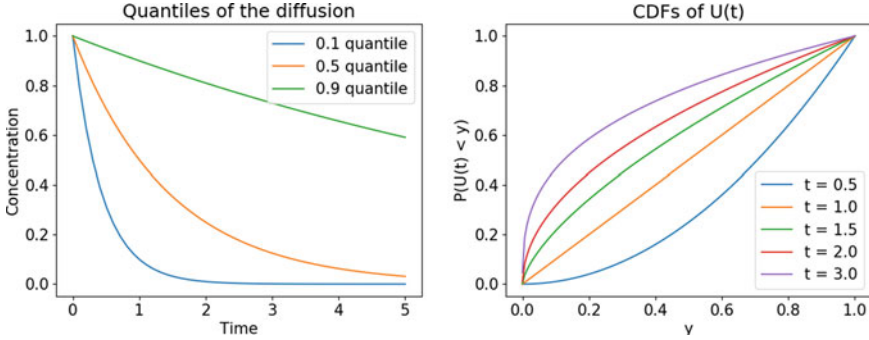


Fig. 2.2 The evolution of quantiles of $U(t)$ (left) and CDFs of $U(t)$ at various times (right)

$$\begin{aligned}
 F_{U(t)}(x) &= Pr(U(t) < x) = P_A(\{a : u(t; a) < x\}) \\
 &= Pr(A > u_t^{-1}(x)) = 1 - F_A(u_t^{-1}(x)) \\
 &= \exp(-\lambda u_t^{-1}(x)) = x^{\frac{\lambda}{t}},
 \end{aligned}$$

where $u_t^{-1}(x) = \frac{\log x}{t}$.

Examples of derived CDFs and evolutions of α -quantiles of $U(t)$, the values x such that $F_{U(t)}(x) = \alpha$, are depicted in Fig. 2.2.

Since we have drifted away from the crisp true–false values of boolean logic towards a multi-valued logic in which each statement is assigned a probability, the *degree of faith*, the meaning of comparative statements becomes unclear (apart from some special cases). Several possible orderings are available for pairs of RVs. We are able to evaluate the probability that they will be ordered once realised, $Pr(X < Y)$. Another widely used one is the *stochastic dominance* for which $X \geq_{st} Y$ iff $\forall x : F_X(x) \leq F_Y(x)$. Nevertheless, given two RVs X, Y , the order among their realisations $x = X(\omega)$, $y = Y(\omega)$ may differ depending on $\omega \in \Omega$, where Ω represents the sample space (Sect. 2.3.1). Generally, there is no unique way of defining which RV is greater in a pair. For us to be able to compare them in an optimisation algorithm, we need to redefine the problem, so that we obtain a total ordering in the range of the objective function (i.e. so that we can compare any pair of proposals by their fitnesses), and so that we uniquely determine whether possible constraints of the problem were violated.

Both may be achieved by replacing the RVs with some meaningful functionals derived from the distribution of the original RV, such as the expected values (Definition 2.4). This approach is justified in mass production scenarios where we try to optimise our long-run (financial) gains according to the law of large numbers (Theorem 2.2). But a different approach should be taken in the case of robust design optimisation, which is often solved by deriving the worst-case scenario as the objective function. A crisp worst case may not be available if the original objective function is a RV, but we can study what is the *likely worst*

scenario, a value which will not be exceeded with *high probability*. This can be achieved by taking some quantile or a risk measure in general (see Chap. 13) or [3, Ch. 12] as the new objective. Similarly, for the case of constrained optimisation, we may, again, demand that the violation will be *unlikely*, i.e. that the probability of violation will be low. This would lead to a redefinition of the constraint as $Pr(\text{constraint is violated}) \leq \alpha$. In both cases, we need to specify a concrete number that represents what exactly are these high and low probabilities, a priori.

In the aforementioned cases, the optimised objective is replaced by a real function, so standard optimization algorithms can be used.

Let us consider that we, again, want to determine the earliest safe time to enter the contaminated area from Example 2.1, now with the uncertain parameter a modelled as a RV with exponential law defined earlier in this section.

The objective function (Eq. (2.3)) is unaffected by the uncertainty in the parameter; thus, we can keep it as it is. The constraints will now have to be reformulated, because $U(t)$ is a RV.

If we admit that we cannot ensure the safety certainly, we can still aim for a *low probability* of encountering the dangerous environment and replace the constraint by bounding the probability of exceeding the safety limits, $Pr(u(t, A) > u_s)$, say by α ($= 0.01, 0.001, 0.0001 \dots$). The optimisation problem derived from Eq. (2.3) will be reformulated as

$$\min_{t \geq 0} t \quad s.t. \quad Pr(U(t) > u_s) \leq \alpha. \quad (2.6)$$

The explicit solution, due to monotonicity, will be attained for the first t for which the $1 - \alpha$ quantile of $U(t)$ will be equal to u_s . Thus, the answer of the model is

- “The smallest safe time t for visiting the area is $\lambda \frac{\ln(u_s)}{\ln(1-\alpha)}$ ”.

Although probability theory provides a convenient modelling framework, it is difficult to properly encode available information into probabilistic models. To construct and manipulate the models, we often have to postulate additional assumptions, which we may not be able to justify (independence of RVs, specific low-dimensional distribution models, etc.). Also, even if our assumptions were correct, if we were to construct the models using the methods of statistical inference, we could only approach the *true* model asymptotically, as the number of samples would approach infinity. But in engineering applications, we often have only a small number of observations, which makes standard inference methods unreliable. Also, the knowledge elicitation process requires that a domain expert exactly assigns probabilities to each possible event, which is generally considered impossible. The situation is, in a sense, analogical to that of providing point estimates, here for the distributions, and may be solved either by modelling the uncertainty by a hierarchical stochastic model, or, again, by introducing imprecision.

2.2.4 A Set of Probability Distributions

In the framework of IP, we will combine the approaches introduced in Sects. 2.2.2 and 2.2.3. The core idea is to consider a set of probability distributions among which we do not make any further judgements about their likeliness of being the *true* model. Including the likeliness, as in the case of hierarchical modelling, would result in a collapse of the set of admissible distributions into a single one, a mixture of distributions. With a purely imprecise model, we are again allowed to ask for expected values and probabilities of events, such as in the precise case, but the answers are now set-valued (often a simple interval). This is analogical to what had happened when we asked for the value of a function with an imprecisely specified parameter in Sect. 2.2.2.

A rigorous approach to IP will be described in the following sections. Let us now, for the sake of an introduction, just consider what would happen if we only knew that the parameter a from the Example 2.1 is an *imprecise* RV A , which follows one of the distributions $\mathbb{P}_A := \{\exp(\lambda) : \lambda \in [\underline{\lambda}, \bar{\lambda}]\}$; however, we cannot further specify which one. We can ask for probabilities of events $A \in E$, derived events $u(t; A) \in E'$ and the expected values. Given that all these depend on the underlying probability distribution, we can, as in Sect. 2.2.2, consider all the possible values they can attain over the set of all the admissible distributions \mathbb{P}_A . For practical reasons, we can just focus on the lower and upper bounds for $P(\cdot)$ and $\mathbb{E}[\cdot]$.

Similarly, as in the case of precise distribution, we would like to assess the CDF of $U(t)$. In the IP framework, and using monotonicity properties of our example, we can construct bounds on the inferred CDF of $U(t)$.

$$\begin{aligned} F(U(t) < x) &\in \left[\min_{p \in \mathbb{P}_A} \{1 - F_p(u_t^{-1}(x))\}, \max_{p \in \mathbb{P}_A} \{1 - F_p(u_t^{-1}(x))\} \right] \\ &= \left[\exp(-\bar{\lambda}u_t^{-1}(x)), \exp(-\underline{\lambda}u_t^{-1}(x)) \right] \\ &= \left[x^{\left(\frac{\bar{\lambda}}{t}\right)}, x^{\left(\frac{\underline{\lambda}}{t}\right)} \right] =: [\underline{P}(U(t) < x), \bar{P}(U(t) < x)], \end{aligned}$$

where \underline{P} and \bar{P} are, respectively, the so-called lower and upper probability measures, which will be properly introduced in Sect. 2.4.

Similarly, we can do with the quantiles. These would again be given as extremes over the set of admissible distributions. An example of such an inference is depicted in Fig. 2.3.

The theory of imprecise probabilities provides a more general theoretical framework for modelling different types of uncertainties and the subsequent (consistent) reasoning. Boolean logic measures statements as true or false, and probabilistic logic measures each statement by a real number, $\cdot \in [0, 1]$, the probability of them being true. The IP framework introduces a possibility for modelling ignorance. For each statement, it can supply a probability of it being true and the probability of the

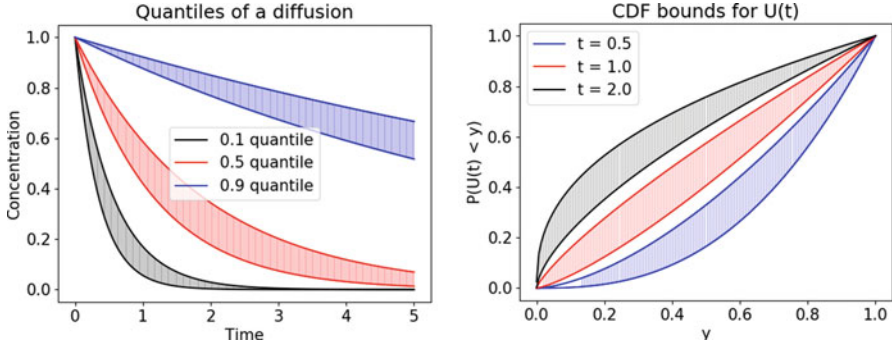


Fig. 2.3 The evolution of the set of various quantiles (left) and the lower and upper CDFs of $U(t)$ for various times (right)

its contrary being true. In the probabilistic logic, these would sum to one due to the axioms of probability measures (Sect. 2.3.1). This is not the case in IP theory. IP assigns to each statement its lower and upper probabilities, infimum and supremum of the probabilities assigned by the distributions in the imprecise model \mathbb{P} . The sum of lower probabilities of a statement and its complement may be lesser than one, and the sum of the upper probabilities may be higher. IP introduces an additional metric for each statement measuring our ignorance—how much we do not know. For a statement E , our ignorance is quantified as $\overline{P}(E) - \underline{P}(E)$ [18]. Note that our ignorance is always zero in the precise probability framework.

For the purposes of design optimisation, we need to combine ideas introduced in Sects. 2.2.2 and 2.2.3. First, the problem needs to be reformulated as it was in the probabilistic scenario, i.e. all the (imprecise) RVs need to be replaced by some functionals. Then, we need to select a way to treat the indeterminacies because these functionals will be interval-valued. Γ -maximin and Γ -maximax may be used, and more decision-making rules can be found in [3, Chap. 8].

Let us return to the optimisation problem given by Eq. (2.6) derived for the stochastic formulation. The objective function will remain real-valued, but the derived constraint violation probability will now be set-valued. Using the Γ -maximin and Γ -maximax approaches yield either of the following:

- “The smallest (reasonably) safe time t for visiting the area is $\overline{\lambda} \frac{\ln(u_s)}{\ln(1-\alpha)}$ ” in the pessimistic (Γ -maximin) case, which guarantees compliance with the weakened (small probability of occurring) form of regulations.
- “The smallest (reasonably) safe time t for visiting the area is $\underline{\lambda} \frac{\ln(u_s)}{\ln(1-\alpha)}$ ” in the optimistic (Γ -maximax) case, which provides solution in which compliance with the weakened regulations is *possible*, but not assured.

2.3 Probability Theory

In this section, we will review selected topics of probability theory. In the two following sections, we will do the same for the theory of imprecise probabilities. Since both these fields cover a vast range of topics, we have decided to focus on what constitutes the underlying structure of these theories and how they relate to each other. Special emphasis will be put on *extending* our (partial) specification of the model to answer enquiries about derived quantities consistently. This means that given some claims about some aspects of some RVs, we are interested in what other claims can be deduced about transformed RVs.

We will show two complementary approaches for building an axiomatic theory of probability. The first one is based on Kolmogorov's formulation [27], in which a probability distribution is represented by a positive additive measure (Sect. 2.3.1). Such measure, which is a set function, directly encodes the modelled probabilities of various assertions about the outcomes of random experiments, allows us to assess an *expected value* of a RV and also extends the models to derived random quantities. This approach to probability theory has become dominant across fields as it offers an intuitive description of random outcomes and enables us to construct efficient general algorithms for solving many practical problems (Monte Carlo algorithms, Bayesian inference, etc.). The measure-theoretic formulation will then be generalised for IP in Sect. 2.4.

Another approach for constructing an axiomatic base for probability theory is based on a functional representation of random quantities [15, 44]. Here, each probability distribution is represented by a functional, the *prevision*, which corresponds to the expected value operator in the measure-theoretic approach. A model is specified by assessing the expected values for several selected functions—the RVs. This allows us to pose less assumptions on the models since the underlying probability measure does not need to be specified exactly, but also pose limitations in extending the assessments to derived quantities. These extensions will (mostly) result only in bounds on the expectations of the derived RVs. The approach will be fully generalised for imprecise probabilities in Sect. 2.5.

2.3.1 Measure-Theoretic Probability

Suppose that we are to perform an experiment. We will denote the set of all its possible outcomes as the *sample space*, Ω . Let us further assume that an outcome cannot be exactly determined prior to its actual observation—it is uncertain. But even though the experimental outcome can be random (i.e. we are not able to predict it using any finite algorithm), multiple repetitions of the same experiment may follow some predictable *law*. Probability theory aims to describe these laws.

Definition 2.1 Let Ω be a sample space and \mathcal{A} a σ -field (a collection of subsets) over Ω . We will call a set function $P : \mathcal{A} \rightarrow \mathbb{R}$ a *probability measure* if:

- $\forall E \in \mathcal{A} : P(E) \in [0, 1]$.
- $P(\Omega) = 1$.
- $\forall E_i \in \mathcal{A}$, which are mutually disjoint, : $P(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i)$.

To distinguish the set function P from the ones which will follow, we will call the tuple $\mathcal{K} := (\Omega, \mathcal{A}, P)$ from Definition 2.1 a (K -)probability field.

Let us again consider the experiment with the set of possible outcomes Ω . If the law of outcomes can be described by a probability distribution P , it indicates that over multiple repetitions of the (exactly the same) experiment, for any chosen $E \in \mathcal{A}$, the relative number of outcomes which will be elements of E , will converge to $P(E)$ as the number of repetitions increases. The distance can be again described probabilistically and will be mentioned again in Theorem 2.3.

We will now introduce the RVs.

Informally, consider a sample space which contains “all the possible states of the universe”, all the possible collapses of wave functions, whimsies of Maxwell demons, of the gods of four winds, etc. Imagine that there exists a measuring device, a different one for each of the quantities of our interest, which is able to read the state of the universe and return the value which our QoI has obtained at the moment. This is how the RVs work.

Definition 2.2 Let (Ω, \mathcal{A}, P) be a K -probability field and $(\Omega_X, \mathcal{A}_X)$ a measurable space. Any \mathcal{A}_X - \mathcal{A} -measurable function $X : \Omega \rightarrow \Omega_X$ will be called a *RV* (RV).

The above definition puts emphasis on the fact that a RV is a function from the sample space—like the measuring device from the informal definition. Now, we are interested in how we can derive the distribution of the RVs. The probability measure from Definition 2.1 calculates the probabilities of events in the sample space. We need to *propagate* this model to assess the statements about RVs. The answer will also allow us to specify the distributions of other derived quantities, such as $Z = f(X, Y)$.

Definition 2.3 For an arbitrary mapping $f : X \rightarrow Y$, for any set $E \subset Y$, we define its *pre-image* as

$$f^{-1}(E) := \{a \in X : f(a) \in E\}.$$

For a set $E \in \mathcal{A}_X$, the probability that the RV X obtains a value in E is

$$P(X \in E) = P(X^{-1}(E)) = P(\{\omega : X(\omega) \in E\}) \stackrel{\Delta}{=} P_X(E). \quad (2.7)$$

Each RV induces its own probability field $(\mathcal{Q}_X, \mathcal{A}_X, P_X)$, where $P_X = P \circ X^{-1}$. We will use notation $X \sim P_X$ for denoting that P_X is the (induced) measure of X . For the derived quantities, say $Z = f(X, Y)$, we may proceed analogically.

Furthermore, the situation is simplified if the sample spaces Ω and Ω_X are the real lines and the mapping defining the new RV is monotone.

Theorem 2.1 *Let $X : (\mathbb{R}, \mathcal{B}, P) \rightarrow (\mathbb{R}, \mathcal{B})$ be a strictly increasing RV and $[a, b] \subset \mathbb{R}$ an interval. Because there exists unique classical inverse X^{-1} of X , which is also an increasing function, we can express the distribution of X as*

$$\begin{aligned} P_X([a, b]) &= P(X \in [a, b]) = P(X < b) - P(X < a) \\ &= P([X^{-1}(a), X^{-1}(b)]). \end{aligned}$$

And similarly for a decreasing X , where the interval for the pre-image is given by swapping the bounds, i.e. $[X^{-1}(b), X^{-1}(a)]$.

See that Theorem 2.1 was exploited in the examples presented in Sects. 2.2.3 and 2.2.4.

In probability theory, special attention is given to the CDFs, which represent the probability that a real-valued RV obtains a value smaller than the argument. A CDF $F_X : \mathbb{R} \rightarrow [0, 1]$ represents, for each $a \in \mathbb{R}$, the probability of event $\{X \in [-\infty, a]\}$. The probability of all the other events in the Borel algebra on the real line, \mathcal{B} , can be derived through the axioms of probability measures. Therefore, CDF uniquely represents the whole probability measure, which would be intractable to work with otherwise.

In special cases, the CDF of a derived quantity can be easily derived from Theorem 2.1. For increasing functions f , the CDF of an extended RV $Y = f(X)$ can be calculated as

$$F_Y(y) = P(f(X) < y) = F_X(f^{-1}(y)).$$

Similarly, for the case of a decreasing function f , where $F_Y(y) = 1 - F_X(f^{-1}(y))$.

Compared with the analysis on the real line, it is convenient to also introduce some summaries of the RVs. This can be done using the *expected value* functional $(\mathbb{E} : \mathcal{L}(\Omega) \rightarrow \mathbb{R})$, where \mathcal{L} is the space of all real RVs on the sample space Ω . For that, we need to equip the underlying probability field with an integration operator (e.g. Lebesgue–Stieltjes integral).

Definition 2.4 Let X be a RV on (Ω, \mathcal{A}, P) . Then,

$$\mathbb{E}_{P_X}[X] := \int_{\Omega_X} x dP_X = \int_{\Omega} X(\omega) dP$$

will be called the *expected value* of RV X . The subscript representing the underlying distribution (P_X) is usually omitted, and we will also do so for precise RVs. We introduce it due to the necessity of computing expected values for various probability measures later in the chapter.

The expected value represents a *typical value*, the limit of average values of multiple draws. By the *law of large numbers*, the average of the finite amount of draws from X will converge towards $\mathbb{E}[X]$.

Theorem 2.2 (Law of Large Numbers) Let X_1, \dots, X_n be a series of RVs, such that each of them is distributed according to the same law with a finite expected value $\mathbb{E}[X_1] =: \mu \in \mathbb{R}$. Then,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n X_i = \mu,$$

where the real number μ can be viewed as a degenerate RV M s.t. $\forall \omega \in \Omega : M(\omega) = \mu$.

The law of large number is the core principle which allows us to perform statistical inference. It guarantees that we will approach correct assessments about the sampling distributions (means, other moments, probability statements, etc.) with increasing number of observations.

Another important result of the probability theory, the *central limit theorem*, states what is the asymptotic convergence rate towards these values. It also provides theoretical guarantee for convergence of Monte Carlo algorithms [28].

Theorem 2.3 (Central Limit Theorem) Let X_1, \dots, X_n be a series of RVs, such that each of them is distributed according to the same law with a finite expected value $\mathbb{E}[X_1] =: \mu \in \mathbb{R}$ and a finite variance $\mathbb{E}[(X_1 - \mu)^2] =: \sigma^2 \in \mathbb{R}^+$. Then

(continued)

Theorem 2.3 (continued)

$$\lim_{n \rightarrow \infty} \frac{1}{\sqrt{n}\sigma} \sum_{i=1}^n (X_i - \mu) \sim \mathcal{N}(0, 1),$$

where $\mathcal{N}(0, 1)$ is the standard normal distribution.

2.3.2 Probability via Expectation

Another approach for building an axiomatic theory of probability is explained by Whittle [44]. The idea is that, instead of focusing on the probability distributions as measures on a sample space, we investigate the RVs from the functional perspective. Whittle therefore describes a system based on axioms posed on the expected values, instead of that based on the probability measures. He further shows how to reproduce results of the standard approach (measure-based) to probability theory.

Definition 2.5 Let (Ω, \mathcal{A}) be a measurable space and $\mathcal{L}(\Omega)$ the set of all RVs on Ω . We will call a functional $\mathbb{E} : \mathcal{L}(\Omega) \rightarrow \mathbb{R}$ the *expected value* if

- $\forall X \in \mathcal{L}(\Omega) : X \geq 0 \Rightarrow \mathbb{E}(X) \geq 0$.
- $\forall a, b \in \mathbb{R}, \forall X, Y \in \mathcal{L}(\Omega) : \mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y)$.
- $\mathbb{E}(\mathbf{1}) = 1$.
- if, $\forall \omega \in \Omega$, a sequence $X_n(\omega)$ increases monotonically to $X(\omega)$, then $\mathbb{E}(X) = \lim \mathbb{E}(X_n)$.

Starting from the axioms of the expected value functional leads to the same theoretical system as Kolmogorov's approach. In the precise probability case, both the approaches are equivalent, but this is no longer the case with imprecise probabilities, where the theory of lower previsions (Sect. 2.5) allows us to represent a larger set of models than the IP extension of the measure-theoretic approach (Sect. 2.3).

A similar approach for formulating probability theory was also explored by de Finetti and Savage [15, 35] from a decision-making perspective. A notable difference introduced by de Finetti is that the expected values are called *previsions* but denote the same object. Also, in the notation, de Finetti further does not distinguish between the symbols for probability of an event, and for an expectation of a RV, both are P , as there exists a one-to-one mapping between events and binary RVs. We can obtain the probability of any event $E \subset \Omega$ by simply calculating the expected value of its indicator function $I_E \in \mathcal{L}(\Omega)$. The meaning is usually evident from the context.

$$I_E(\omega) := \begin{cases} 1, & \omega \in E, \\ 0, & \omega \notin E, \end{cases} \quad (2.8)$$

$$P(E) = \mathbb{E}(I_E).$$

The prevision terminology was also adapted in the lower prevision theory.

An interesting feature, that both Whittle and de Finetti have presented, is the possibility to extend our partial knowledge to other RVs. Given a set of known expectations for RVs $\mathcal{K} \subset \mathcal{L}(\Omega)$, we can derive bounds for the expectation of another RV $Y \notin \mathcal{K}$. The procedure is called simply an *extension*.

To do this, we can use corollaries of the expectation axioms. Given two RVs (mappings from the sample space) s.t. $\forall \omega \in \Omega : X(\omega) \leq Y(\omega)$, we can derive, due to the linearity and positivity of the expectation functional, that $\mathbb{E}(X) \leq \mathbb{E}(Y)$.

Therefore, if we already know the expectations $\mathbb{E}(X_n)$ of several RVs X_n , we trivially know the expectation of their arbitrary countable linear combination $Z = \sum a_i X_i$, which is $\mathbb{E}(Z) = \sum a_i \mathbb{E}(X_i)$.

For an arbitrary RV Y , we can obtain the lower bound on its expectation by taking the supremum over all the RVs in the span of X_1, \dots, X_n , which are strictly lower than Y . Similarly for the upper bound.

Theorem 2.4 *Let $\mathcal{K} = \{X_1, \dots\}$ be a countably infinite set of RVs with known expectations and $\mathcal{Z} := \{\sum_{i=1}^n a_i X_i + b : X_i \in \mathcal{K}, a_i, b \in \mathbb{R}, n \in \mathbb{N}\}$ the linear span of $\mathcal{K} \cup \{\mathbf{1}\}$. The consistent (coherent in de Finetti's treatment) bounds for the expected value $\mathbb{E}(Y)$ can be obtained as*

$$\sup_{Z \in \mathcal{Z}, Z \leq Y} \mathbb{E}(Z) \leq \mathbb{E}(Y) \leq \inf_{Z \in \mathcal{Z}, Z \geq Y} \mathbb{E}(Z),$$

where by $Z \geq Y$, we mean that $\forall \omega \in \Omega : Z(\omega) \geq Y(\omega)$.

For Ω, \mathcal{K} , which are both finite, where $|\Omega| = N, |\mathcal{K}| = n$, the extension is a linear program.

$$\sup_{Z \in \mathcal{Z}, Z \leq Y} \mathbb{E}(Z) = \sup_{\substack{b \in \mathbb{R}; \\ \forall \omega \in \Omega: b + \sum_{i=1}^n a_i X_i(\omega) \leq Y(\omega)}} \sup_{a \in \mathbb{R}^n} b + \sum_i^n a_i \mathbb{E}(X_i).$$

Its dual is

$$\sup_{Z \in \mathcal{Z}, Z \leq Y} \mathbb{E}(Z) = \inf_{\substack{p \in (\mathbb{R}_0^+)^n \\ \sum_{i=1}^n p_i = 1 \\ \forall X_j \in \mathcal{K}: \sum_{i=1}^n p_i X_j(\omega) = \mathbb{E}(X_j)}} \sum_i^n p_i Y(\omega_i),$$

which effectively means that we are extremising the expectation over some *set of admissible distributions* p , which would yield the known expectations for all $X_n \in \mathcal{K}$.

2.4 Imprecise Probabilities

Compared with the point-valued and imprecise specification of the parameter in Sects. 2.2.1 and 2.2.2, we take the approach for generalising the precise probability theory by considering sets of precise distribution models for a RV. Analyses in the IP framework are then conducted by analysing the properties of these sets. But specifying such sets and working with them are not always as simple as in Sect. 2.2.4.

In the rest of the chapter, we are going to introduce the underlying mathematical structures needed for the treatment of imprecisely specified RVs.

2.4.1 A Set of Measures

We begin by introducing an extension to measure-theoretic probability given by Weichselberger [43] and similarly by Walley [41]. The idea is to assign to every event $E \in \mathcal{A}$ a pair of real numbers $\in [0, 1]$, which represent the lower and upper bounds for the probability of that event. These bounds will reoccur within the rest of the theory of imprecise probabilities. They also have an epistemological interpretation—the lower bound measures the evidence supporting the occurrence of E , whereas the upper bound measures the evidence, which contradicts E . Their difference is the measure of our ignorance.

Definition 2.6 An interval-valued set function P on a measurable space (Ω, \mathcal{A}) will be called a *R-probability* if:

- $\forall E \in \mathcal{A} : P(E) = [L(E), U(E)]$ s.t. $0 \leq L(E) \leq U(E) \leq 1$.
- The set $\mathcal{M} := \{p : p \text{ is K-probability}, \forall E \in \mathcal{A} : p(E) \in P(E)\} \neq \emptyset$.

(continued)

Definition 2.6 (continued)

We will call the tuple $\mathcal{R} := (\Omega, \mathcal{A}, L, U)$ a *R-probability field*. The set \mathcal{M} from the second axiom is called the *structure* (Weichselberger) or the *credal set* (Walley) of \mathcal{R} .

The letter R- indicates *reasonable*. It corresponds to the property of *avoiding sure loss* introduced in Sect. 2.5. The definition directly implies that for an R-probability P , $L(\emptyset) = 0$ and $U(\Omega) = 1$ through the non-emptiness of the credal set.

An R-probability is directly connected to a set of precise probabilities via its credal set. It is also apparent that for any set of precise probabilities, we may construct an R-probability s.t. this set will be a subset of its credal set, but such construction may not be unique. To specify an R-probability, we would need to define the L and R functions for all the elements in the respective algebra \mathcal{A} , which is quite impractical.

Simpler IP models can be used to assess judgements about more complex ones. If we specify an R-probability via a (possibly finite) set of precise models, we automatically include all their convex combinations in the structure of such an R-probability. Given two K-probabilities $p, q \in \mathcal{M}$, the credal set of some R-probability, for all their convex combinations $r = \lambda p + (1 - \lambda)q$, $\lambda \in [0, 1]$, $r(E)$ will take values in between $p(E), q(E)$ for all the events $E \in \mathcal{A}$ due to the axioms of probability measures. Thus, $L(E) \leq r(E) \leq U(E)$, so $r \in \mathcal{M}$.

Therefore, the credal set of an R-probability is equal to its convex hull.

A one-to-one correspondence between probability bounds and the underlying credal set is a desired feature in IP theory. In Weichselberger's treatment, this may be achieved by tightening the requirements on the probability bounds L, U .

Definition 2.7 An R-probability P , which also satisfies

$$\forall E \in \mathcal{A} : \left(\inf_{p \in \mathcal{M}} p(E) = L(E) \right) \quad \wedge \quad \left(\sup_{p \in \mathcal{M}} p(E) = U(E) \right),$$

is called an *F-probability* and the corresponding tuple $\mathcal{F} := (\Omega, \mathcal{A}, L, U)$, the *F-probability field*.

The letter F- indicates *feasible*. It corresponds to the notion of *coherence* introduced in Sect. 2.5. For every F-probability, we also immediately imply that $L(\Omega) = 1$ and $U(\emptyset) = 0$, as this is true for each of the elements of the credal set, therefore also their infimum and supremum. The axioms of F-probabilities directly imply a relation which is reoccurring throughout many parts of IP theory and which

Table 2.1 Example of F- and R-probabilities on a simple finite sample space $\Omega = \{0, 1, 2\}$

E	L_F	U_F	P	L_R	U_R
\emptyset	0.00	0.00	0.00	0.00	0.33
$\{0\}$	0.33	0.60	0.50	0.33	0.60
$\{1\}$	0.30	0.50	0.40	0.30	0.50
$\{2\}$	0.00	0.37	0.10	0.00	0.55
$\{0, 1\}$	0.63	1.00	0.90	0.00	1.00
$\{0, 2\}$	0.50	0.70	0.60	0.00	0.90
$\{1, 2\}$	0.40	0.67	0.50	0.00	0.83
$\{0, 1, 2\}$	1.00	1.00	1.00	0.50	1.00

enables us to focus our attention solely on either the L or U function. The other follows through the conjugacy property:

$$\forall E \in \mathcal{A} : U(E) + L(E^c) = 1. \quad (2.9)$$

Since an F-probability defines an underlying credal set and the probability assessments can be obtained through extremisation over this set, we may also define the lower and upper expected values for RVs through similar extremisation. The lower expectation would be given by Eq. (2.10) and the upper one similarly by taking a supremum instead of the infimum.

$$\underline{\mathbb{E}}[X] := \inf_{p \in \mathcal{M}} \mathbb{E}_p[X]. \quad (2.10)$$

Example 2.2 An example of F- and R-probabilities is presented in Table 2.1. L_F, U_F and L_R, U_R correspond to F- and R-probability bounds, respectively. A K-probability is also shown in column P to demonstrate non-emptiness of the respective credal sets.

A desirable property of the IP framework is the possibility to derive probability bounds on all the events $E \in \mathcal{A}$ from the knowledge of the bounds only for some events $E' \in \mathcal{A}' \subset \mathcal{A}$. This operation is commonly referred to as an *extension* (with some adjectives corresponding to the theories and actual definitions).

Definition 2.8 Let (Ω, \mathcal{A}) be a measurable space. Denote $\mathcal{A}' = \mathcal{A} \setminus \{\emptyset, \Omega\}$, and let $\mathcal{A}_L, \mathcal{A}_U \subset \mathcal{A}'$.

If there exists a non-empty set \mathcal{M} of probability distributions and set functions L, U s.t.

- $\forall E \in \mathcal{A}_L, p \in \mathcal{M} : L(E) \leq p(E)$,

(continued)

Definition 2.8 (continued)

- $\forall E \in \mathcal{A}_U, p \in \mathcal{M} : U(E) \geq p(E),$
- $L(\emptyset) = U(\emptyset) = 0, \quad L(\Omega) = U(\Omega) = 1,$

then $P = (L, U)$ is called a *partially determinate R-probability*.

We will call $(\mathcal{A}_L, \mathcal{A}_U)$ the *support* of P .

Definition 2.9 Let $(\Omega, \mathcal{A}, L, U)$ be a partially determinate R-probability field with support $(\mathcal{A}_L, \mathcal{A}_U)$. If also

- $\forall E \in \mathcal{A}_L : L(E) = \inf_{p \in \mathcal{M}} p(E),$
- $\forall E \in \mathcal{A}_U : U(E) = \sup_{p \in \mathcal{M}} p(E),$

then P is called *partially determinate F-probability*.

For partially determinate F-probabilities, there exists a straightforward way of calculating probability bounds for events outside of their support. Such procedure is called *normal completion* in Weichselberger's and *natural extension* in Walley's treatment. It simply exploits the extremising property of F-probabilities over their respective credal sets \mathcal{M} , thus

$$\forall E \in \mathcal{A} : L(E) = \inf_{p \in \mathcal{M}} p(E). \quad (2.11)$$

Example 2.3 Let us consider a partially determinate F-probability on $\Omega = \{0, 1, 2\}$, $\mathcal{A}_L = \mathcal{A}_U = \{\{0\}, \{1\}\}$ with L, U on \mathcal{A}_L given in Table 2.1, and a credal set \mathcal{M} .

For an arbitrary event $E \in \mathcal{A}$, we can calculate its lower probability by solving the optimisation problem

$$L(E) = \min_{p \in \mathcal{M}} p(E).$$

Especially, denoting $a := P(\{0\}), b := P(\{1\}), c := P(\{2\}),$

$$\begin{aligned} L(\{0, 1\}) &= \min_{\substack{0.33 \leq a \leq 0.67 \\ 0.1 \leq b \leq 0.17 \\ a + b + c = 1}} a + b = 0.67. \end{aligned}$$

The bounds for the rest of the events in \mathcal{A} are given in Table 2.1.

Weichselberger's treatment provides a useful framework to define the lower and upper probabilities and even the desired extending properties for partial specifications. Probability bounds given for all the elementary events $E \in \mathcal{A}' \subset \Omega$ can be extended into bounds for arbitrary event $E \in \Omega$ by solving a linear optimisation problem.

For the derived RVs $Y = f(X)$, we may calculate the imprecise probabilities that they will obtain a value in an element of their respective σ -algebras, similarly as in the precise case through Eq. (2.7) and Theorem 2.1. The imprecision in the distributions will also manifest in the imprecision in the expected values. We can calculate the lower and upper expectations, but for the structures introduced in this section, we can only do so through by optimising the expected value over the credal set.

2.4.2 Capacities

Capacities originated in the work of Choquet [10] on the generalisation of measure theory for non-additive measures. They further provide useful properties and structure to IP models, which allows us to simplify the specification of the bounds for arbitrary event $E \in \mathcal{A}$ and the expectations of RVs.

Definition 2.10 Let (Ω, \mathcal{A}) be a measurable space. A set function $g : \mathcal{A} \rightarrow \mathbb{R}$ is called a *capacity* if it is monotone, i.e.

$$\forall A, B \in \mathcal{A} : A \subset B \Rightarrow g(A) \leq g(B).$$

A capacity g is further called *super-additive* if

$$\forall A, B \in \mathcal{A} : A \cap B = \emptyset \Rightarrow g(A \cup B) \geq g(A) + g(B).$$

If the inequality is reversed, it is instead called *sub-additive*.

Note that if the structure \mathcal{M} of an F-probability is closed (i.e. arginf belongs to \mathcal{M}), then both L and U are super- and sub-additive capacities, respectively.

Definition 2.11 A capacity g is said to be *n-monotone* if for any collection $E_n \subset \mathcal{A}$ of n elements

(continued)

Definition 2.11 (continued)

$$g\left(\bigcup_{E \in E_n} E\right) \geq \sum_{\mathcal{E} \subset E_n} (-1)^{|\mathcal{E}|+1} g\left(\bigcup_{E \in \mathcal{E}} E\right).$$

If g is monotone for every $n \in \mathbb{N}$, then it is called ∞ -monotone.

Corollary

Any n -monotone capacity is also $n > m$ -monotone.

Two-monotone capacities are coherent. A pair of super- and sub-additive capacities, such that the sub-additive one dominates the super-additive one, constitute an F-probability.

Definition 2.12 For a super-additive capacity g defined on a *finite* space Ω , we define, for every event $E \subset \Omega$, a function $m_g : 2^\Omega \rightarrow \mathbb{R}$, the *möbius inverse*, as

$$m_g(E) := \sum_{A \subset E} (-1)^{|E \setminus A|} g(A).$$

The benefit is that an inverse mapping exists, which enables us to reconstruct the capacity from its möbius inverse as

$$g(E) = \sum_{A \subset E} m_g(A). \quad (2.12)$$

The dual capacity, the upper probability, can be reconstructed as

$$g^*(E) = \sum_{\{A \in 2^\Omega : A \cap E \neq \emptyset\}} m_g(A). \quad (2.13)$$

A special class of models is composed of ∞ -monotone lower probabilities on finite spaces. Their möbius inverses (aka the *mass functions*) are non-negative for every event. Conversely, any normalised ($\sum m = 1$) non-negative function $m : 2^\Omega \rightarrow \mathbb{R}$ with a finite support induces ∞ -monotone lower and upper probabilities by Eqs. (2.12) and (2.13), respectively. This is explicitly exploited in the evidence theory (see Chap. 17).

Example 2.4 Let us assume that we have a collection of open-interval-valued measurements: $\{(0.2, 0.6), (0.4, 0.8), (0.1, 0.3)\}$.

By the Laplace indifference principle, we assign to each of the interval an equal mass $m = \frac{1}{3}$. With such a mass function, we can construct lower and upper probabilities via Eqs. (2.12) and (2.13). For example,

$$L([0.3, 1]) = m((0.4, 0.8)) = \frac{1}{3}$$

$$U([0.3, 1]) = m((0.2, 0.6)) + m((0.4, 0.8)) = \frac{2}{3}.$$

Capacities also provide us means for calculating the bounds not only on probabilities of events but also on the expected values of RVs. If an F-probability is viewed as a pair of capacities, we can define an integration functional, which will enable us to compute the bounds on the expected value, similar to which we are used to from the precise probability theory (Definition 2.4).

Definition 2.13 For a capacity $g : \mathcal{A} \rightarrow \mathbb{R}$ and a real-valued function f measurable on \mathcal{A} , the *Choquet integral* is defined as

$$(C) \int f dg = \int_0^\infty g(f \geq x) dx + \int_{-\infty}^0 (g(f \geq x) - 1) dx,$$

where the integrals on the right-hand side are Riemann's and $f \geq x$ denotes $\{t \in \Omega : f(t) \geq x\}$.

Theorem 2.5 For a coherent 2-monotone lower probability $g : \mathcal{A} \rightarrow \mathbb{R}$, the lower expectation of a function f is given by the Choquet integral.

$$\underline{\mathbb{E}}(f) = (C) \int f dg.$$

Remark: if the capacity represents a 2-monotone upper probability, the integration would yield the upper expectation.

2.4.3 Neighbourhood Models

One simple way of defining an IP model is by taking a neighbourhood of some precise one. In the settings of optimisation under uncertainty, such models would provide a straightforward way for imposing robustness of the results. Such models may also be directly used for sensitivity analysis and for the analysis of the robustness of statistical procedures [25].

There are several ways to construct an IP from a baseline precise distribution P_0 [3, Ch. 4]. The *Pari-Mutuel* model enables us to directly evaluate the lower and upper probabilities it encodes. It originated with the intention to ensure bookmakers a positive expected gains from a series of lotteries (i.e. buying tickets cheaper than their known expected gains and selling them for more). For a fixed baseline P_0 and a contamination parameter ϵ , the lower and upper probabilities of events are defined as

$$\underline{P}(E) = \max\{(1 + \epsilon)P_0(E) - \epsilon, 0\}, \quad \overline{P}(E) = \min\{(1 + \epsilon)P_0(E), 1\}.$$

Another widely used model is the *linear-vacuous* model, which is directly defined for the lower and upper expected values (also for probabilities using Eq. (2.8)). The idea is to construct a mixture of the baseline model P_0 and a *vacuous* model which assigns the infimum and supremum values of a function as their lower and upper expectations. For a contamination parameter ϵ and RV $X \in \mathcal{L}(\Omega)$,

$$\underline{\mathbb{E}}[X] = (1 - \epsilon)\mathbb{E}_{P_0}[X] + \epsilon \inf_{\Omega} X(\omega)$$

From the IP point of view, the lower and upper probabilities induced by the *Pari-Mutuel* model are 2-monotone, and those from the *linear-vacuous* one are ∞ -monotone capacities. Both models are therefore coherent. Therefore, for the *Pari-Mutuel*, the bounds on expectations can be computed using the Choquet integral (Theorem 2.5).

2.4.4 Random Sets

A set-valued evidence may be encountered in numerous practical scenarios, may it be the error bounds of measuring devices or an interval-valued expert elicitation. There exists an approach for handling these within precise probability theory in the case that we know that the imprecision is not inherent to the actual realisation of the experiment and only comes as a *coarsening* of precise values via our imperfect methods. In such a case, we may introduce an additional assumption on the stochastic nature of how the coarsening occurs, a conditional model on where

the actual value lies in the set (e.g. as in the treatment of censored data in reliability theory Chap. 4). But in some cases, this assumption may be unjustifiable and bias our assessments. The imprecision may also be caused by the very nature of the of the experiment, where the random observation itself is set-valued and cannot be treated using the mentioned method. To rigorously address these situations, probability theory may be generalised for the set-valued observations into the theory of random sets [30, 31, 33].

Definition 2.14 Let (Ω, \mathcal{A}, P) be a probability space, \mathcal{S} a collection of subsets of Ω_Φ and $\Phi : \Omega \rightarrow \mathcal{S}$ a map.

If

$$\{\omega : \Phi(\omega) \cap E \neq \emptyset\} \in \mathcal{A}; \quad \forall \text{ compact } K \subset \Omega_\Phi,$$

then we will call Φ a *random set*.

The definition of a random set is almost identical to that of a RV (Definition 2.2). We just need to impose proper measurability properties. Nevertheless, the treatment of the random sets is slightly different. For a random set Φ , we can assess several claims.

Definition 2.15 Let $\Phi : \Omega \rightarrow \mathcal{S} \subset 2^{\Omega_\Phi}$ be a random set derived from probability space (Ω, \mathcal{A}, P) . Then, for $E \in \mathcal{S}$ and $x \in \Omega_\Phi$, we define the following:

The *belief function* $Bel(E) := P(\Phi \subset E)$.

The *plausibility function* $Pl(E) := P(\Phi \cap E \neq \emptyset)$.

The *contour function* $C(x) := P(x \in \Phi)$.

The belief and plausibility functions corresponds to lower and upper probabilities and L and R functions from Sect. 2.3, respectively.

Random set theory is a basis for the Dempster–Shafer theory of evidence, described in Chap. 17 and some modern statistical methods [29]. Random set models have also been used for sensitivity analysis [34] and uncertainty modelling in general [22]. As IP models, the belief and plausibility functions induced by random sets are ∞ -monotone capacities and, therefore, coherent lower probabilities. Therefore, we also know the form for the derived lower and upper expectations via Theorem 2.5.

Random sets can be used for statistical inference with little assumptions. The models can (and have been [34]) be constructed from Chebyshev’s inequality (Theorem 2.6) if only the population mean and variance are known. This represents

the tightest bounds for the respective probabilities over all the possible probability distributions compliant to these assumptions. If the population mean and variance are unknown, Saw [36] has proposed a variant of the Chebyshev’s inequality based on their sample estimates.

The construction from Chebyshev’s inequality defines a random set induced by a uniformly distributed RV, say $U \sim Uni([0, 1])$. The random sets, as models for a RV with mean μ and variance σ^2 , are constructed via mapping $\Theta_C(u) := [\mu - \frac{\sigma}{\sqrt{u}}, \mu + \frac{\sigma}{\sqrt{u}}]$. An example of such a constructed random set is depicted in Fig. 2.4.

Theorem 2.6 (Chebyshev’s Inequality) For a RV X with finite expectation $\mu = \mathbb{E}(X)$ and finite non-zero variance $\sigma^2 = \mathbb{E}((X - \mu)^2)$ and $\forall a \in \mathbb{R} : a > 0$,

$$P(|X - \mu| \geq a\sigma) \leq \frac{1}{a^2}.$$

Since they are analogical to precise probability theory, some of the useful results are also available for the theory of random sets. Especially variants of the law of large numbers (Theorem 2.2) and the central limit theorem (Theorem 2.3) can be generalised for random sets [31]. Alvarez [1] and Balch [4] provide means on how to conduct Monte Carlo simulation using random set models by constructing an *empirical* random set from drawn samples, such that the approximations of the *Bel* and *Pl* functions are unbiased estimates and converge almost surely to the population ones.

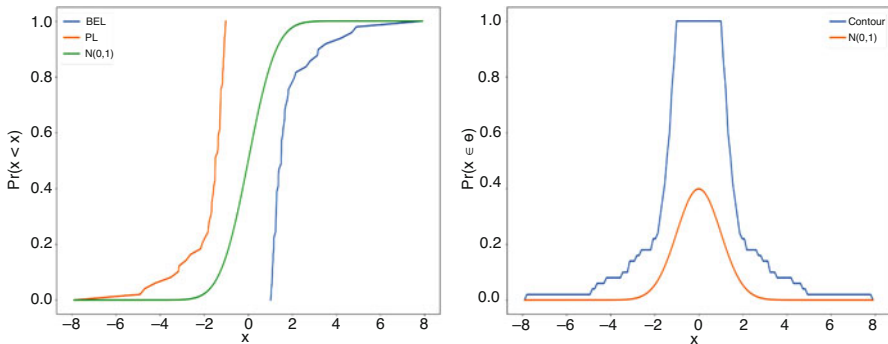


Fig. 2.4 An example of a random set constructed from Chebyshev’s inequality with $\mu = 0, \sigma = 1$. We compare its belief and plausibility function with the CDF of the standard normal distribution (left) and its contour function with the PDF of the standard normal distribution (right)

2.4.5 Probability Boxes

A probability box (P-box) is defined by two cumulative distribution functions $\bar{F} \geq \underline{F}$ [20]. The set of probability distributions, which a P-box envelopes, is composed of all the CDFs which are bounded by \underline{F} and \bar{F} . The resulting lower probability \underline{P} is ∞ -monotone and coherent. In the case that an additional knowledge is available, additional bounds may also be imposed, e.g. on the mean values and variances of the enveloped distributions to further narrow the structure. Standard arithmetical operations can be generalised for P-boxes so they can be used for risk and sensitivity analysis, similarly to the random sets.

In the multivariate case, multiple assumptions about the correlation of RVs can be imposed, including the situation when the correlation is regarded as entirely unknown [21]. An example of inferences from different dependency assumptions is shown in Fig. 2.5 for the addition of two RVs with a uniform distribution (any precise distribution is also a P-box with a single element in its credal set). So, P-boxes allow us to refrain entirely from specifying any assumption about dependency. The calculation is based on the Frechet inequalities [21].

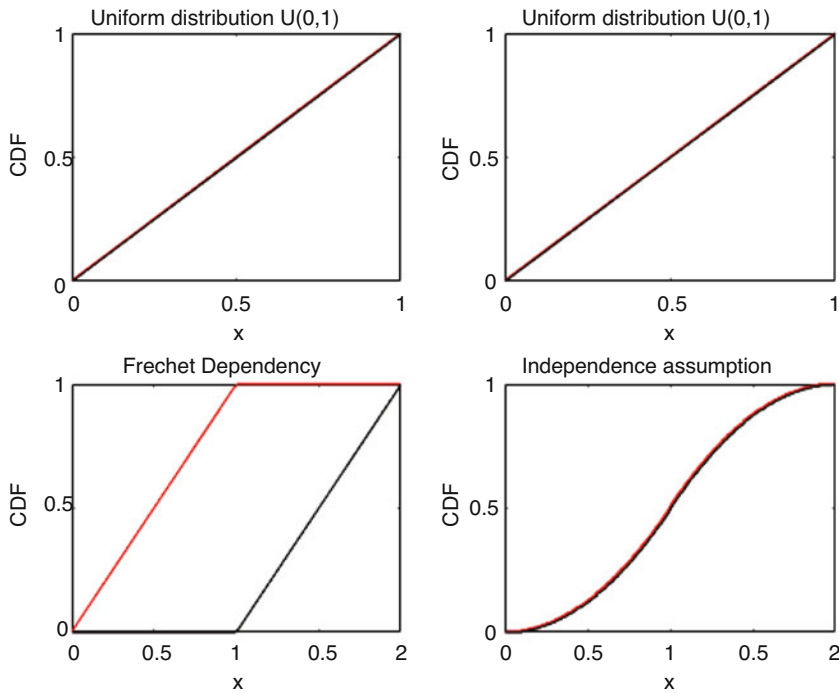


Fig. 2.5 P-boxes for the sum of two uniform RVs. P-boxes for the uniform RVs are shown at the top portion, whereas the P-boxes of their sum based on different dependency assumptions are presented at the bottom part: no assumption (left) and independence assumption (right)

2.5 Lower Previsions

The theory of previsions starts with the work of de Finetti [14, 15]. Later generalisation of prevision to lower prevision occurred in Williams' work [45, 46] and in a more developed form in Walley's work [40]. This is another approach of IP. The theory of lower previsions has a unifying mathematical character that attracted considerable attention. There are several other concepts like probability charges (Bhaskara Rao, Bhaskara Rao [7]), n -monotone set functions (Choquet [10]), belief functions (Dempster [16], Shafer [37]), possibility measures (De Cooman [12]), and many others (De Cooman [13], Denneberg [19], Troffaes [38]). They can also be regarded as lower expectations with respect to closed convex sets of probability measures.

In this section, we will start with the notion of desirability and then derive the main theory behind lower previsions. Finally, we will conclude this with a discussion of duality between lower previsions and credal sets.

2.5.1 Desirability

Suppose we observe an experiment where we have different outcomes and we are betting on these outcomes for some rewards. Gambles can be seen as these uncertain rewards on the set of possible outcomes, say Ω . Mathematically, a gamble f is a real-valued bounded function on Ω . The set of all gambles is denoted by \mathbb{B} . We can add or subtract two gambles as usual.

Example 2.5 Imagine, we are watching a horse race involving three horses: A , B and C . Therefore, we have the set of outcomes $\Omega = \{A, B, C\}$, where the elements denote the events where the corresponding horse wins. The dealer is offering several gambles on the race. For example, if we consider the gamble g_1 , we win 3 if A wins and 5 if B wins. However, we lose 1 if C wins. We have shown a list of available gambles in Table 2.2.

(continued)

Table 2.2 Gambles for betting

	A	B	C
g_1	3	5	-1
g_2	2	-1	5
g_3	0	0	-1
g_4	4	-2	10
g_5	1	4	0
g_6	4	9	-1
g_7	6	5	5

Example 2.5 (continued)

We also have additional information that C got injured before the race and A won the previous race. However, we have no information for B .

2.5.1.1 Axioms

Definition 2.16 (Desirability) We call a gamble desirable if we accept it depending on the available information about the outcomes.

In our example in Table 2.2, we can see that for g_1 , we will lose 1 if C wins. However, based on our information, C is injured and less likely to win the race; therefore, we can accept this gamble. Conversely, we see that for g_2 , we can lose some amount if B wins. But because A won the previous race, we may accept this gamble.

Besides accepting g_1 and g_2 , we accept or reject a gamble depending on some rationality criteria. We can see that $g_3 < 0$, that is, we can't win anything, and eventually, we may lose some amount if C wins the race. Therefore, we will not choose this gamble as we cannot win anything; i.e. we want to avoid sure loss.

Here, by $g < 0$, we mean that $g(x) \leq 0 \quad \forall x \in \Omega$ and that there exists at least one $x \in \Omega$, such that $g(x) < 0$.

Contrarily, for g_5 , we won't be losing anything, and we can actually win some if A or B wins the race. Therefore, g_5 is a safe bet for gambling, and we will accept g_5 ; i.e. we want to accept sure gain.

Now, from the example, it can be also seen that $g_4 = 2g_2$ and $g_6 = g_1 + g_5$. Therefore, if we are committed to accepting g_2 , then we should also accept g_4 ; similarly, if we are committed to accepting g_1 and g_5 , then we should also accept g_6 .

We can see the abovementioned rationality criteria as desirability axioms for gambling. We list these axioms as follows:

1. **Avoiding sure loss:** For any gamble g , if $g < 0$, then we don't accept g .
2. **Accepting sure gain:** For any gamble g , if $g \geq 0$, then we accept g .
3. **Positive homogeneity:** If we accept a gamble g , then we accept λg for any $\lambda \geq 0$.
4. **Combination:** If we accept two gambles, g and g' , then we will accept $g + g'$.

From the abovementioned axioms, we can prove the following statement: If we accept g and $g' \geq g$, then we accept g' . That is, if g' is a bounded gamble that dominates the accepted bounded gamble g , then we accept the bounded gamble g' too. We can view this as a *monotonicity* property of desirability [39].

2.5.2 Lower Previsions

From our example, we see that the dealer is offering five different gambles, which are listed in Table 2.2. Now, we are willing to buy g_1 from the dealer. We are already informed that C is injured, so the chance of winning for C is less than that of A . Therefore, we can pay a higher price than the reward on C , as we expect C not to win. However, for B , we are not sure, as we have no information about B . Therefore, we will not spend more than three, as otherwise, if A wins, then we will end up losing some amount. However, even if B wins, then we will earn some reward. We show the transactions in the following Table 2.3.

Our supremum buying price 3 for g_1 can be seen as our lower prevision for g_1 . We can formulate this behavioural interpretation of lower previsions as follows:

Definition 2.17 (Lower Prevision) Lower prevision $\underline{P}(g)$ of a gamble can be seen as the supremum buying price of the gamble. Or in other words, $\underline{P}(g)$ is the lowest value, such that we are willing to buy the gamble for all $t < \underline{P}(g)$.

Similarly, we associate another map $\overline{P}(g)$ called *upper prevision* from the set of gambles to the real numbers. Here, upper prevision stands for infimum selling price. That is, if we own a gamble g , then we can sell the gamble for all $t > \overline{P}(g)$.

For any bounded gamble g , the following conjugacy property holds:

$$\overline{P}(g) = -\underline{P}(-g) \tag{2.14}$$

This allows us to express one type of functional in terms of the other [39].

Suppose after buying g_1 , we decide to buy another gamble g_2 for 3. Then, we can see our total rewards in the following table.

We can see from Table 2.4 that we will end up losing some amount irrespective of the outcome. Therefore, we incur a sure loss.

Definition 2.18 (Avoiding Sure Loss) A lower prevision is said to avoid sure loss if for every gamble g_1, g_2, \dots, g_n and for all non-negative real numbers

(continued)

Table 2.3 Betting on g_1

	A	B	C
Reward on g_1	3	5	-1
Buy g_1 for 3	0	2	-4
Buy g_1 for 5	-2	0	-6

Table 2.4 Betting on g_1 and g_4

	A	B	C
Buy g_1 for 3	0	2	-4
Buy g_2 for 3	-1	-4	2
Total reward	-1	-2	-2

Definition 2.18 (continued)
 $\lambda_1, \lambda_2, \dots, \lambda_n$, the following relation holds:

$$\sup \sum_{i=1}^n \lambda_i [g_i - \underline{P}(g_i)] \geq 0 \tag{2.15}$$

Avoiding sure loss can be derived directly from the desirability axioms. Here, the $[g_i - \underline{P}(g_i)]$'s are desirable gambles because of the interpretation of the lower prevision. The $\lambda_i [g_i - \underline{P}(g_i)]$'s are desirable because of the positive homogeneity. The $\sum_{i=1}^n \lambda_i [g_i - \underline{P}(g_i)]$ is desirable because of the combination of desirable gambles. Finally, the supremum comes from avoiding sure loss.

Now, if we buy g_2 for 1, then we have the following transaction:

Clearly, for this specific consideration, we no longer incur sure loss with the revised buying price for g_2 . Avoiding sure loss for the above example can be verified as follows:

$$\max_{\omega \in \Omega} \lambda_1 [g_1(\omega) - 3] + \lambda_2 [g_2(\omega) - 1] \geq 0 \tag{2.16}$$

for all $\lambda_1, \lambda_2 \geq 0$. We will verify this later after introducing duality in Sect. 2.5.4.

2.5.3 Natural Extension

In between the horse race, the dealer offers a new gamble f on the winner of the race:

	A	B	C
g	5	4	4

Clearly, we can see that $g \geq g_1 + g_2$. Therefore, by the monotonicity property, we can buy this gamble. Since $g \geq g_1 + g_2$, we can buy it for at least $\underline{P}(g_1) + \underline{P}(g_2) = 3 + 1 = 4$.

But 4 is not necessarily the maximum buying price for g . This leads to the idea of natural extension, wherein we try to assess a new gamble outside the domain based on our assessment of the gambles inside.

Definition 2.19 (Natural Extension) Let \underline{P} be a lower prevision and $g_i \in \text{dom}\underline{P}$ for $i = 1, 2, \dots, n$; then, for any gamble g , we can define the natural extension \underline{E} of \underline{P} as follows:

$$\underline{E}(g) = \sup \left\{ a \in \mathbb{R} : g - a \geq \sum_{i=1}^n \lambda_i [g_i - \underline{P}(g_i)], n \in \mathbb{N}, \lambda_i \in \mathbb{R}_{\geq 0} \right\} \quad (2.17)$$

The natural extension in Eq. (2.17) can be derived directly from the desirability axioms. The $\lambda_i [g_i - \underline{P}(g_i)]$'s are desirable because of the positive homogeneity. The $\sum_{i=1}^n \lambda_i [g_i - \underline{P}(g_i)]$ is desirable because of the combination of desirable gambles. The term $g - a$ is desirable because of the monotonicity of desirable gambles.

Definition 2.20 (Coherence) A lower prevision, \underline{P} , is called coherent if

$$\underline{P}(g) = \underline{E}(g) \quad (2.18)$$

for all $g \in \text{dom}\underline{P}$.

Coherence means that our supremum buying price of a gamble should not be raised on the combination of other gambles. For example, in Table 2.2, we can see that $g_7 \geq g_1 + g_2$. Therefore, we can dispose to buy g_7 for 4; it also avoids sure loss. However, clearly, we can buy this gamble for 5 without any loss. This is against the assumption that 4 is the supremum buying price for g_7 ; therefore, it leads to inconsistency.

2.5.4 Duality

In the previous section, we derived the natural extension \underline{E} of \underline{P} . Now, for a finite number of gambles g_1, g_2, \dots, g_n and finite set of outcomes $\Omega \equiv \{x_1, x_2, \dots, x_k\}$, we can write Eq. (2.17) in the following manner:

$$\begin{aligned}
& \max_{a \in \mathbb{R}} a \\
& \text{subject to,} \\
& a + \sum_{i=1}^n \lambda_i [g_i(x_j) - \underline{P}(g_i)] \leq g(x_j) \\
& \lambda_i \geq 0 \quad i = 1, 2, \dots, n
\end{aligned} \tag{2.19}$$

for $j = 1, 2, \dots, k$. So in this way, we can see it as an optimisation problem, with a being the objective function. Then, writing Eq. (2.19) in matrix form, we get the following linear programming problem:

$$\begin{aligned}
& \max_{\mathbf{v} \in \mathbb{R} \times \mathbb{R}_{\geq 0}^n} \mathbf{c}^T \mathbf{v} \\
& \text{subject to,} \quad \mathbf{A} \mathbf{v} \leq \mathbf{b}
\end{aligned} \tag{2.20}$$

where

$$\mathbf{v} := \begin{bmatrix} a \\ \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} \quad \mathbf{c} := \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{A} := \begin{bmatrix} 1 & [g_1(x_1) - \underline{P}(g_1)] \cdots [g_n(x_1) - \underline{P}(g_n)] \\ 1 & [g_1(x_2) - \underline{P}(g_1)] \cdots [g_n(x_2) - \underline{P}(g_n)] \\ & \vdots \\ 1 & [g_1(x_k) - \underline{P}(g_1)] \cdots [g_n(x_k) - \underline{P}(g_n)] \end{bmatrix}$$

$$\text{and } \mathbf{b} := \begin{bmatrix} g(x_1) \\ g(x_2) \\ \vdots \\ g(x_k) \end{bmatrix}$$

Then, by the duality principle, we can get following dual of Eq. (2.20):

$$\begin{aligned}
& \min \sum_{j=1}^k p_j g(x_j) \\
& \text{subject to} \\
& \sum_{j=1}^k p_j g_i(x_j) \geq \underline{P}(g_i) \\
& \sum_{j=1}^k p_j = 1 \\
& p_j \geq 0 \\
& i = 1, 2, \dots, n \quad j = 1, 2, \dots, k
\end{aligned} \tag{2.21}$$

Remark Here, the equality in the dual model occurs as a is a free variable in the original problem.

Here, p_j 's ($j = 1, 2, \dots, k$) are the probability mass functions. Note that $\sum_{j=1}^k p_j g(x_j)$ is the expectation of gamble g with respect to the probability mass functions p_1, p_2, \dots, p_n . The feasible region forms a convex set of probability mass functions. We call this convex set of probability mass functions the *credal set*. This duality relation shows the interesting connection between the lower prevision and credal sets \mathcal{M} . It also shows that the natural extension of a lower prevision is the lower expectation with respect to the credal set.

Theorem 2.7 (Lower Envelope Theorem) *Let \underline{P} be any lower prevision, and let \mathcal{M} be the corresponding credal set. Then, the following statements are true:*

1. \underline{P} avoids sure loss if \mathcal{M} is non-empty.
2. If \underline{P} avoids sure loss, and then, its natural extension \underline{E} is the lower envelope of \mathcal{M} , that is, it satisfies

$$\underline{E}(g) = \min \left\{ \sum_{j=1}^k p_j g(x_j) : p_j \in \mathcal{M} \quad j = 1, 2, \dots, k \right\} \text{ for all } g \in \mathbb{B} \quad (2.22)$$

3. \underline{P} is coherent if it avoids sure loss and for all $g \in \text{dom } \underline{P}$

$$\underline{P}(g) = \min \left\{ \sum_{j=1}^k p_j g(x_j) : p_j \in \mathcal{M} \quad j = 1, 2, \dots, k \right\} \quad (2.23)$$

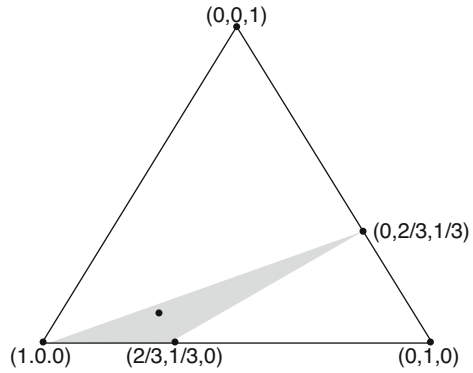
The above theorem is a direct consequence of the duality relation derived from the natural extension for a finite set of outcomes and a finite domain of \underline{P} . It shows that the natural extension of the lower previsions is the lower expectation. This can also be extended for infinite sets, and the generalisation can be proved using the Hahn–Banach theorem. This allows us to characterise these notions of avoiding sure loss, coherence and natural extension of the lower prevision in terms of their dual models [40].

We can derive this condition for avoiding sure loss using the example in Table 2.5. We saw in the example that, we no longer incur sure loss with the revised buying price. This can be verified with the help of Eq. (2.16). We can write Eq. (2.16) as follows:

Table 2.5 Revised betting on g_1 and g_4

	A	B	C
Buy g_1 for 3	0	2	-4
Buy g_2 for 1	1	-2	4
Total reward	1	0	0

Fig. 2.6 Avoiding sure loss



$$\begin{aligned}
 & \max_{a \in \mathbb{R}} \quad -a \\
 & \text{subject to,} \\
 & \lambda_1 [g_1(x_j) - 3] + \lambda_2 [g_2(x_j) - 0] - a \leq 0
 \end{aligned} \tag{2.24}$$

for $j = 1, 2, 3$. Then, by taking the dual of the above problem, we get the following set of conditions:

$$\begin{aligned}
 & \min \quad 0 \\
 & \text{subject to,} \\
 & \sum_{j=1}^3 p_j g_i(x_j) \geq \underline{P}(g_i) \\
 & \sum_{j=1}^3 p_j = 1 \\
 & p_1, p_2, p_3 \geq 0
 \end{aligned} \tag{2.25}$$

where $\underline{P}(g_1) = 3$ and $\underline{P}(g_2) = 1$. For instance, $p = (2/3, 1/4, 1/12)$ satisfies all the constraints and hence avoids sure loss.

In Fig. 2.6, we see that with the revised buying price, we avoid sure loss within the area shaded by grey. Here, the triplet $(p_1, p_2 \text{ and } p_3)$ stands for the probability of winning of the horses A, B and C respectively. The black dot within the feasible region is $p = (2/3, 1/4, 1/12)$. This shows that if we avoid sure loss, then there

exists $p_j \in \mathcal{M}$, such that our expected reward on each gamble is higher than the supremum buying price of each gamble. That is, \mathcal{M} is non-empty.

2.6 Constructing the Laws

The classical interpretation of probability is the relative frequency of occurrence. Approximate models of the underlying laws may be constructed from finite number of observations as probability distributions. Conversely, the subjectivistic point of view does not treat probabilities as objective quantities but rather as a tool to describe our state of knowledge about quantities and our degrees of belief in statements. The problem of what probability is and how it can be measured is more rigorously addressed in [14, 32]. Statistical methods are usually employed for model construction if observations are available. These seek a way to find a mathematical form of the probability distribution of the uncertain quantity which complies with the available information. If observations are not present, we need to rely on an expert opinion to construct the models via the process of expert knowledge elicitation. We will omit the expert elicitation process in this chapter. Some ideas of what can be elicited in the subjective setting can be found in [15] for precise models or in [40] for IP models.

In this section, we will briefly remind the basics of statistical inference. For an exhaustive treatment, we refer the reader to [9]. We are going to revise the basic principles to extract knowledge from the available data. Our main aim was to introduce how the inferential procedures can be extended for the IP theory.

2.6.1 Statistical Inference with Precise Probabilities

Let us now recall the basic methods of statistical inference for data analysis. Hereafter, we will assume that we have a set of measurements $\vec{x} = \{x_1, \dots, x_n\}$, independent and identically distributed (i.i.d.) samples, which were generated according to some precise *ground-truth* distribution \hat{P} . Our final intention was to provide probabilities of various events of interest based/conditioned on this dataset. The common practice is to construct an approximation (model) P of the sampling distribution \hat{P} and estimate the desired probabilities from P .

A simple way of inferring probability distributions from a set of samples is given by *non-parametric methods*. Here, for an arbitrary event E , the probability is estimated as $P(E) = \frac{1}{n} \#\{x_i; x_i \in E\}$, the relative ratio of observations which comply with E . Distributions inferred in this way are usually labelled as empirical and constitute models with least additional assumptions. An example of an empirical CDF is depicted in Fig. 2.7 with a label “empirical”.

Alternatives to the non-parametric methods search for an approximative distribution by inverting the model of the sampling process. They look for an answer

in an apriori-selected set of candidate distributions, say $\mathbb{P} := \{P_\theta, \theta \in \Theta\}$. Based on the axiomatic theory of probability, precise methods mainly comprise of two competing methodologies—frequentist and Bayesian. Nevertheless, the common inference scheme for constructing distributional point estimates, i.e. selecting a single *best-fitting* probability distribution, is simply as follows:

1. Choose (subjectively) a set of plausible sampling distributions \mathbb{P} .
2. Construct the *likelihood function* $\mathcal{L}(\theta; \vec{x})$, which models the probability of observing the collection \vec{x} for each parameter θ .
3. Select $\hat{\theta}$ that best fits the observations and approximate \hat{P} by $P = P_{\hat{\theta}}$ (the frequentist approach), or construct a mixture of distributions from the chosen family with mixing weights $w(\theta) \propto \mathcal{L}(\theta; \vec{x})\pi_0(\theta)$ and approximate \hat{P} by $P = w(\theta)P_\theta$ (the Bayesian approach; π_0 is called the prior distribution).
4. Evaluate the approximations of the desired probabilities from inferred distribution P .

The samples may come in various forms. Most commonly, they are considered precisely specified (e.g. real values for a real RV), in which case the likelihood function for inference from a set of independent samples will take the form

$$\mathcal{L}(\theta; \vec{x}) = \prod_{i=1}^n f_\theta(x_i), \quad (2.26)$$

where f_θ is the probability density function of distributions from the chosen family \mathbb{P} indexed by θ .

Example 2.6 Let us assume that we have a set of observations $\vec{x} := \{x_1, \dots, x_N\}$ of a positive RV X . We choose the set of admissible sampling distributions of the RV X to be the set of all exponential distributions ($F(x; \theta) = 1 - \exp(-\theta x)$). The frequently used frequentist method is the so-called *maximum-likelihood estimate* (MLE). Here, we seek such value of θ which maximises the likelihood function (Eq. (2.26)). Thus,

$$\theta_{MLE} = \operatorname{argmax}_{\theta \in \Theta} \mathcal{L}(\theta; \vec{x}),$$

and construct $P = P_{\theta_{MLE}}$.

The resulting CDF is depicted in Fig. 2.7 with a label “MLE”.

Example 2.7 Let us assume the same scenario as in Example 2.6. We again select the set of admissible sampling distributions of the RV X , \mathbb{P} , to be

(continued)

Example 2.7 (continued)

the set of all exponential distributions. But now, we will employ a Bayesian procedure where we model our *knowledge* about distribution parameter θ by a probability distribution instead of selecting just one *best-fit value*.

First, a *prior* distribution $\pi_0(d\lambda)$, which represents our knowledge about λ before observing the data, has to be elicited. Then, our knowledge is refined via the Bayes updating rule to construct our *posterior* knowledge about θ as

$$w(\theta) = p(\theta|\vec{x}) \propto \mathcal{L}(\theta; \vec{x})p_0(\theta). \tag{2.27}$$

Equation (2.27) specifies the posterior probability density function (the mixing weight) up to a normalisation constant. From that, we can construct the predictive distribution for a future sample X_{n+1} as a weighted average of predictions of all the models in \mathbb{P} . Thus,

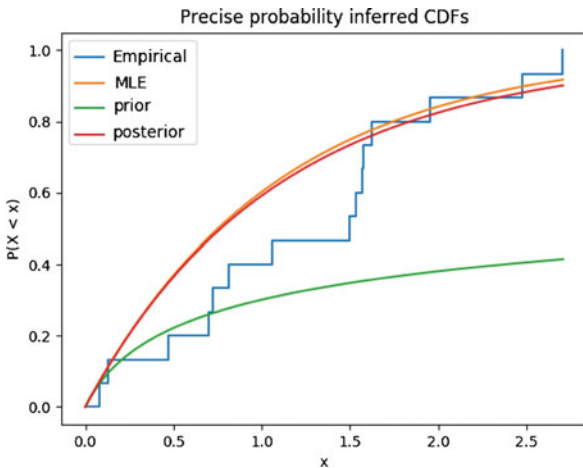
$$p(x_{n+1}|\vec{x}) = \frac{\int_{\theta} p(x_{n+1}|\theta)p(\theta|\vec{x})d\theta}{Z(\vec{x})}, \tag{2.28}$$

where $Z(\vec{x})$ is a normalisation constant.

An example of the Bayesian inference is depicted in Fig. 2.7 with CDFs labelled as “prior” and “posterior” for prior and posterior predictive distributions, respectively.

For particular choices of families of likelihood functions, we can find a family of prior distributions which is closed under Bayes’ updating. This means that the posterior distribution lies in the same family, so we only need to update its

Fig. 2.7 Example of precise probability inferences: an empirical distribution, a maximum-likelihood estimate and prior and posterior predictive distributions from the Bayesian inference



parameters. We call these the *conjugated families*. For a particular choice of the exponential model for the observed samples in Example 2.7, the conjugated family of distributions of θ is the gamma distributions. This class also induces a closed form for the posterior predictive distribution (Eq. (2.28)), the Lomax distribution. If no conjugate form can be found for the Bayesian inference, the problem needs to be solved numerically, generally using Monte Carlo algorithms [28].

2.6.2 Robust Bayesian Inference

One application of the IP theory was to provide means for sensitivity analysis for various decision-making problems under uncertainty. In the case of Bayesian inference, it was labelled *robust Bayesian analysis* [6]. In the Bayesian framework, we can analyse the sensitivity on both the prior distribution and/or the observation model, the likelihood function. A straightforward solution is to consider sets of functions (priors and likelihoods) instead of just a single one in the analysis. The set of prior distributions would define an F-probability with the respective credal set. The credal set for the posterior F-probability would be given by the set of all updated priors. All the assertions of interested would then be given by extremisation over the posterior credal set (Eq. (2.10)).

Example 2.8 Assume the same observations as in Examples 2.6 and 2.7 and the same set of admissible sampling models, \mathbb{P} . This leads to the same likelihood function (Eq. (2.26)). Now, assume that we cannot properly specify one prior distribution for the Bayesian analysis as in Example 2.7. Instead, let us consider a set of prior distributions, again conjugated with our likelihood (i.e. gamma distributions).

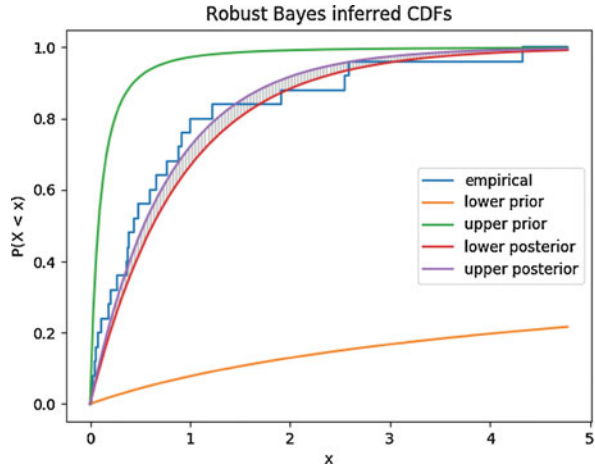
As emphasised in Sect. 2.2.4, while consulting the IP model, we need to consider answers from all the singular models in the credal set. In the case of reconstructing the predictive CDFs, we construct the bounds for all the CDFs from the set of all the updated prior distributions. Therefore,

$$\underline{F}(x) := \min_{\pi \in \Pi_0} \frac{1}{Z_{\pi}(\vec{x})} \int_0^x \int_{\theta} p(x_{n+1}|\theta) \mathcal{L}(\theta; \vec{x}) \pi(\theta) d\theta dx_{n+1}.$$

An example of a robust Bayesian inference is depicted in Fig. 2.8, where the lower and upper bounds are given for both the prior and posterior predictive distributions.

A powerful result of the robust Bayesian inference is a closed-form solution for the *imprecise Dirichlet model* [40]. The power lies in two features. First, the model is constructed for multinomial sample distributions. It can therefore be used

Fig. 2.8 Bounds for prior and posterior predictive CDFs resulting from robust Bayesian inference compared with the empirical distribution of samples



for any inferential scenario with a finite set of outcomes, also for inferences in general spaces after a finite grouping of its elements. The inferred parameters are the probability masses for the considered categories. The second feature is that the imprecise Dirichlet model can model entirely vacuous prior previsions. This means that the prior F-probability for an observation to be of arbitrary category is $(0, 1)$. Compare this with Example 2.8, where even though we have used a set of prior distributions, the prior previsions of $P(X < 4)$ would be approx. $(0.2, 1) \neq (0, 1)$. Imprecise Dirichlet model employs the ideal *non-informative* prior for Bayesian inference, which cannot be modelled by any precise probability distribution.

2.6.3 Frequentist Inference with Imprecise Probabilities

Sensitivity analysis, which is similar to the robust Bayesian statistics, was also explored in the frequentist framework. Frequentist induction, with precise probabilities, assumes that there exists a precise sampling distribution from which the i.i.d. observations are generated. This assumption may be weakened, as has been done by [42], who argue for the possibility of imprecise sampling distributions and describe desirable properties of imprecise frequentist inference. The strong motivation for this extension is the theoretical impossibility of observing *identically* distributed samples due to variability in the experimental setting (although it may be negligible). Their approach includes precise distributions as a special case, and in the case of a precise sampling process, the imprecisions in the inferred distributions converge to zero—the inferred IP law converges to a precise law.

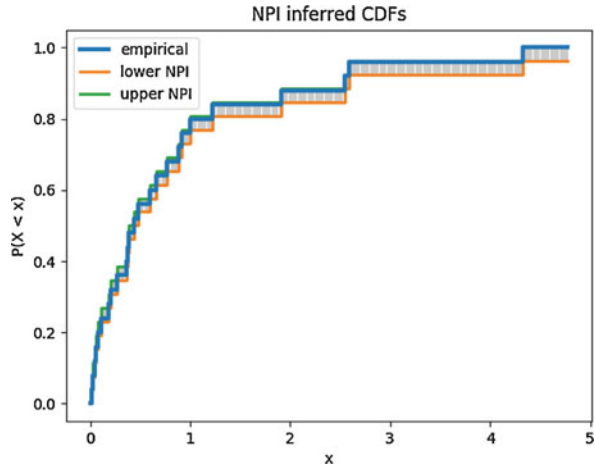
Another line of work in the frequentist inference focuses on the foundations of statistical inference itself. The core notion of frequentist inference lies in the construction of statistical procedures with guaranteed qualitative properties, such as

bounding the type I error in hypothesis testing procedures. Bayesian procedures can only comply with these asymptotically and can be severely biased by the information supplied through the prior distribution in cases when only a small number of observations is available. Conversely, the results of Bayesian procedures, the posterior distributions, can be propagated in a straightforward manner to obtain assertions about the derived quantities, $f(X)$. The problem of obtaining similar distributional estimate in the frequentist framework was studied by Fisher in his work on *fiducial inference* [23]. It later inspired the development of the well-known theory of confidence intervals and hypothesis tests. Nevertheless, fiducial inference has suffered from various, justified drawbacks and has mostly been forgotten by mainstream statisticians. Several attempts had been made on the revival of ideas, and it seems that in order to do so, the inferential results have to be modelled by IP distributions instead of precise ones.

The recent advancements were enabled by Dempster [16, 17] by his development of the evidence theory and its application to statistical problems. In [29], a statistical framework for constructing random set structures, which can be used to obtain valid confidence intervals on any level of significance and to conduct hypothesis tests, is presented. The results of these inferences are generally random sets, beliefs and plausibility functions which can be used to bound the inferences about the investigated RV. The observations are modelled via a pivotal model, where we assume that the observed value is a deterministic function of some ancillary RVs with known distribution, thus extending [24, 47], who aimed at constructing precise frequentist distributional estimates. Ryan et al. used this pivotal relation to propagate a random set prediction of the ancillary RV, by which they obtained statistical procedures with superior properties. Aside from that, they allow us to develop derived methods for situations with additional knowledge, propagate the resulting random sets to obtain assessments about derived quantities, and naturally analyse imprecise observations without additional modelling assumptions.

As another example of a frequentist inferential method, we would like to present the non-parametric predictive inference (NPI) [11]. The method assumes exchangeability of the observations and bases its indifference principle on Hill's assumption—indifference among all possible orderings. Formally, after the n real-valued observations $\vec{x} = \{x_1, \dots, x_n\}$, the NPI constructs a predictive random set for the next observation X_{n+1} by placing mass $1/(n+1)$ on each of the intervals (x_i, x_{i+1}) , $i = 0 \dots n$, where x_0, x_{n+1} are some bounds of the X_{n+1} support (possibly infinite). The corresponding lower and upper probabilities may be derived by Eqs. (2.12) and (2.13). It was shown that the result of the inference is an ∞ -monotone capacity and an F-probability [2]. The NPI has also been extended to include situations with censored observations without the need of including any restrictive censoring assumptions. An example of NPI lower and upper CDFs is shown in Fig. 2.9 compared with the empirical distribution.

Fig. 2.9 Lower and upper cumulative distribution function obtained via NPI



2.7 Concluding Remarks

We have attempted to provide a concise introduction in the theory and methods of imprecise probabilities, although numerous interesting topics have been omitted to keep the text bounded. For a technical overview of the topics of IP theory, we refer the reader to [3], which includes a wide collection of results, detailed explanations of the underlying mathematical structures, examples of practical application, and further references. In addition, a concise overview of the applications of IP in engineering context is given by Beer et al. in [5].

We have included the most fundamental (from our point of view) technical structures, which underlie IP models and subsequent analyses in Sects. 2.4 and 2.5, together with some, hopefully, illuminating examples.

For the purposes of actually working with IP models, the last section was dedicated to demonstrate how these models can be constructed using the methods of statistical inference.

References

1. D.A. Alvarez, On the calculation of the bounds of probability of events using infinite random sets. *Int. J. Approx. Reason.* **43** 241–267 (2006)
2. T. Augustin, F.P.A. Coolen, Nonparametric predictive inference and interval probability. *J. Stat. Plan. Inference* **124**, 251–272 (2004)
3. T. Augustin et al. (eds.), *Introduction to Imprecise Probabilities* (Wiley, New York, 2014), p. 432
4. M.S. Balch, Methods for rigorous uncertainty quantification with application to a Mars atmosphere model. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, 2010

5. M. Beer, S. Ferson, V. Kreinovich, Imprecise probabilities in engineering analyses. *Mech. Syst. Signal Process.* **37**, 4–29 (2013)
6. J.O. Berger et al., An overview of robust bayesian analysis. *Test* **3**, 5–124 (1994)
7. K.P.S. Bhaskara Rao, M.B. Rao, *Theory of Charges: A Study of Finitely Additive Measures*. Pure and Applied Mathematics (Elsevier Science, Amsterdam, 1983)
8. G. Boole, *An Investigation of the Laws of Thought: On Which Are Founded Mathematical Theories of Logic and Probabilities* (Dover, New York, 1854)
9. G. Casella, R.L. Berger, *Statistical Inference* (Thomson Learning, Pacific Grove, 2002, 2010)
10. G. Choquet, Theory of capacities: research on modern potential theory and Dirichlet problem. Technical note, University of Kansas, Dept. of Mathematics, 1954
11. F.P.A. Coolen, Low structure imprecise predictive inference for Bayes' problem. *Stat. Probab. Lett.* **36**, 349–357 (1998)
12. G. de Cooman, Possibility theory I: the measure- and integral-theoretic groundwork. *Int. J. Gen. Syst.* **25**, 291–323 (1997)
13. G. de Cooman, M.C.M. Troffaes, E. Miranda, A unifying approach to integration for bounded positive charges. *J. Math. Anal. Appl.* **340**, 982–999 (2008)
14. B. de Finetti, La Prévision: Ses Lois Logiques, Ses Sources Subjectives. *Ann. Inst. Henri Poincaré* **17**, 1–68 (1937)
15. B. de Finetti, *Theory of Probability: A Critical Introductory Treatment* (Wiley, New York, 2017)
16. A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **38**, 325–339 (1967)
17. A.P. Dempster, New methods for reasoning towards posterior distributions based on sample data. *Ann. Math. Stat.* **37**, 355–374 (1996)
18. A.P. Dempster, The Dempster–Shafer calculus for statisticians. *Int. J. Approx. Reason.* **48**, 365–377 (2008)
19. D. Denneberg, *Non-Additive Measure and Integral* (Springer, Dordrecht, 1994)
20. S. Ferson et al., Constructing probability boxes and Dempster–Shafer structures. Tech. rep., Sandia National Laboratories, 2003
21. S. Ferson et al., Dependence in probabilistic modeling, Dempster–Shafer theory, and probability bounds analysis. Tech. rep., Sandia National Laboratories, 2004
22. T. Fetz, M. Oberguggenberger, Propagation of uncertainty through multivariate functions in the framework of sets of probability measures. *Reliab. Eng. Syst. Saf.* **85**, 73–87 (2004)
23. R.A. Fisher, Inverse probability. *Math. Proc. Camb. Philos. Soc.* **26**, 528–535 (1930)
24. D.A.S. Fraser, *The Structure of Inference* (Wiley, New York, 1968)
25. P.J. Huber, E.M. Ronchetti, *Robust Statistics* (Wiley, New York, 2009)
26. E.T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, 2003)
27. A.N. Kolmogorov, *Foundations of the Theory of Probability* (AMS Chelsea Publication, New York, 1956)
28. D.P. Kroese, T. Taimre, Z.I. Botev, *Handbook of Monte Carlo Methods* (Wiley, New York, 2011)
29. R. Martin, C. Liu, *Inferential Models: Reasoning with Uncertainty* (Chapman and Hall/CRC, Boca Raton, 2015)
30. G. Matheron, *Random Sets and Integral Geometry* (Wiley, New York, 1975)
31. I. Molchanov, *Theory of Random Sets* (Springer, London, 2005)
32. R.F. Nau, De Finetti was right: probability does not exist. *Theory Decis.* **51**, 89–124 (2001)
33. H.T. Nguyen, *An Introduction to Random Sets* (Chapman and Hall/CRC, Boca Raton, 2006)
34. M. Oberguggenberger, W. Fellin, Reliability bounds through random sets: non-parametric methods and geotechnical applications. *Comput. Struct.* **86**, 1093–1101 (2008)
35. L.J. Savage, *The Foundations of Statistics* (Dover, New York, 2012)
36. J.G. Saw, M.C.K. Yang, T.C. Mo, Chebyshev inequality with estimated mean and variance. *Am. Stat.* **38**, 130–132 (1984)
37. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)

38. M. Troffaes, Optimality, uncertainty, and dynamic programming with lower previsions, Jan 2005
39. M.C.M. Troffaes, G. de Cooman, *Lower Previsions* (Wiley, New York, 2014), pp. 37–75
40. P. Walley, *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability (Taylor & Francis, London, 1991)
41. P. Walley, Towards a unified theory of imprecise probability. *Int. J. Approx. Reason.* **24**, 125–148 (2000)
42. P. Walley, T.L. Fine, Towards a frequentist theory of upper and lower probability. *Ann. Stat.* **10**, 741–761 (1982)
43. K. Weichselberger, The theory of interval-probability as a unifying concept for uncertainty. *Int. J. Approx. Reason.* **24**, 149–170 (2000)
44. P. Whittle, *Probability via Expectation* (Springer, New York, 1992)
45. P.M. Williams, Indeterminate probabilities, in *Formal Methods in the Methodology of Empirical Sciences: Proceedings of the Conference for Formal Methods in the Methodology of Empirical Sciences, Warsaw, 17–21 June 1974*, ed. by M. Przełęcki et al. (Springer, Dordrecht, 1976), pp. 229–246
46. P.M. Williams, Notes on conditional previsions. *Int. J. Approx. Reason.* **44**, 366–383 (2007)
47. M.g. Xie, K. Singh, Confidence distribution, the frequentist distribution estimator of a parameter: a review. *Int. Stat. Rev.* **81**, 3–39 (2013)

Chapter 3

Uncertainty Quantification in Lasso-Type Regularization Problems



Tathagata Basu, Jochen Einbeck, and Matthias C. M. Troffaes

Abstract Regularization techniques, which sit at the interface of statistical modeling and machine learning, are often used in the engineering or other applied sciences to tackle high dimensional regression (type) problems. While a number of regularization methods are commonly used, the ‘Least Absolute Shrinkage and Selection Operator’ or simply LASSO is popular because of its efficient variable selection property. This property of the LASSO helps to deal with problems where the number of predictors is larger than the total number of observations, as it shrinks the coefficients of non-important parameters to zero. In this chapter, both frequentist and Bayesian approaches for the LASSO are discussed, with particular attention to the problem of uncertainty quantification of regression parameters. For the frequentist approach, we discuss a refit technique as well as the classical bootstrap method, and for the Bayesian method, we make use of the equivalent LASSO formulation using a Laplace prior on the model parameters.

Keywords Statistical modeling · LASSO · Bayesian statistics · Uncertainty quantification

3.1 Introduction

Statistics is a collection of mathematical concepts to analyze and find the structure in data. Data can be either numeric- or character-valued (representing a class) depending on the problem. There are several purposes of statistics; however one of the main purposes is description of the data and prediction of system behavior from the observed data. Elements of statistical reasoning have been traced back as early as 400 AD [14, p. 7] in India. However, the modern-day approach only started

T. Basu (✉) · J. Einbeck · M. C. M. Troffaes
Durham University, Durham, UK
e-mail: tathagata.basu@durham.ac.uk; jochen.einbeck@durham.ac.uk;
matthias.troffaes@durham.ac.uk

emerging in the eighteenth century, following advances in the theory of probability [14, p. 176].

In this chapter, we will discuss statistical regularization and uncertainty quantification problems using Least Absolute Shrinkage and Selection Operator (LASSO) estimators [24, 25]. The LASSO estimator is a popular regularization method due to its variable selection property. After Tibshirani introduced LASSO in 1996 [24], numerous authors contributed further to the theory, including Osborne, Presnell, and Turlach [21] and Efron et al. [6]. Friedman et al. [10] discussed computational aspects of the LASSO. Park and Casella [22] introduced the Bayesian approach for LASSO estimators, using a hierarchical mixture model for parameter estimation. Other notable works deal with the specification of shrinkage parameter by Lykou and Ntzoufras [18], the Dirichlet LASSO by Das and Sobel [4], and the spike and slab LASSO by Ročková [23].

First, we will introduce the basic notions behind statistical modeling and regularization. In Sect. 3.2, we will look at some important concepts of parameter estimation with and without regularization. Eventually, we will introduce the LASSO estimators in Sect. 3.3. In Sect. 3.4, we will discuss different uncertainty quantification methods for the LASSO followed by an extension to the logistic model in Sect. 3.5. Section 3.6 concludes the chapter.

3.1.1 Statistical Modeling

To make statistical inferences from data, first, we need variables and a model describing the relations between those variables. We can categorize variables into *response variables* and *predictor variables*:

1. Predictor (or independent) variables are characteristics of the system which directly control the properties of the system.
2. Response (or dependent) variables are characteristics of the system which depend on the predictor variables. In other words, they respond to a change of values of the predictors in some systematic fashion.

Assume we have a dataset containing n independent and identically distributed (i.i.d.) observations of real-valued responses $y_1, \dots, y_n \in \mathbb{R}$, along with corresponding vector-valued predictors $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^p$. We consider each \mathbf{x}_i to be a column vector.

Example 3.1 (Gaia Dataset) Gaia is a mission by the European Space Agency (ESA) to formulate a three-dimensional map of our galaxy [8]. The data depicted in Fig. 3.1 are part of a dataset which was simulated prior to the launch of the mission from computer experiments [1, 7]. The data contain essentially spectral information divided into $p = 16$ wavelength bands (intervals), along with certain stellar parameters which are to be inferred from the spectral data. That is, each observation in the data set represents a stellar object, and the measurement for

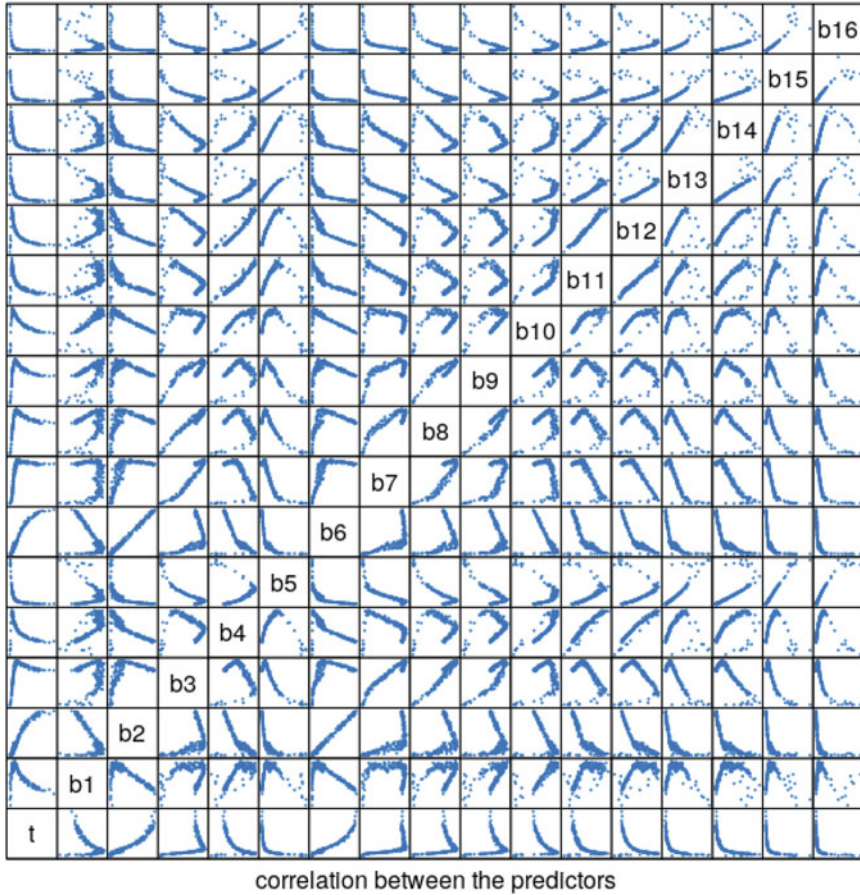


Fig. 3.1 Scatter plot matrix of the Gaia dataset. The variable denoted t (temperature) corresponds to the response; the variables denoted $b1$ to $b16$ (bands) correspond to the predictors. Note that the plot is symmetric w.r.t. the counterdiagonal

each “band” is the energy flux (photon counts) emitted from that object within that wavelength interval.

In this example, stellar temperature (in Kelvin scale) is the response variable. In the dataset that we have available, a total of $n = 8286$ observations (stellar objects) are recorded. It can be seen from Fig. 3.1 that the 16 predictor variables are strongly correlated with each other, suggesting that they carry redundant information.

Often, one of the objectives of statistical modeling is to identify a functional relationship (“model”) between the responses and the predictor variables:

$$E(y_i | \mathbf{x}_i) = \phi(\mathbf{x}_i, \boldsymbol{\beta}) \tag{3.1}$$

where ϕ is a function that depends on a parameter vector β . For instance, as will be described in Sect. 3.1.3 in more detail, in a linear regression context, one typically has $\phi(\mathbf{x}_i, \beta) = \mathbf{x}_i^T \beta$. There also exist non-parametric approaches which do not assume an explicit parametric shape, but most of such approaches achieve this by simply introducing a large number of parameters, so that they still can be expressed as in Eq. (3.1).

3.1.2 Statistical Inference

Statistical inference is the process by which we use the available data to gain knowledge about the model parameters, such as β in Eq. (3.1), as well as their uncertainties. In a wider sense, it will also include methods by which we quantify and validate our assumptions on the model. Statistical inference deals with the estimation of parameters that are used to specify the family of probability distributions which underlie the statistical model for $y_i | x_i$. Inference has several applications in science and engineering. Generally, there are two conceptually different approaches to statistical inference: the *frequentist* approach and the *Bayesian* approach. There are some other concepts available which are beyond the scope of this chapter but are addressed in other articles in this volume.

The frequentist approach is the most widely used estimation method. Sometimes it is referred to as the “classical” approach. The estimation can be a point estimate where we simply try to find the best guess for the parameter of the parametric model. Alternatively, we seek an interval which covers the unknown parameter value with high probability (generally 0.95). We call this a 95% confidence interval.

While several point estimators are available, the *maximum likelihood estimator* (or MLE) is among the most popular because of its simple and wide implementability and its consistency properties. It finds the parameter value which maximizes the probability density of the sample given the parameter, i.e., the likelihood. For linear regression models under normal errors, MLE is equivalent to the ordinary least squares (OLS).

The Bayesian approach starts from Bayes’ rule for conditional probability. Denote the data by \mathbf{Y} . For example, in our setting, \mathbf{Y} is simply the vector of observed response values $(y_1, \dots, y_n)^T$. The statistical model is specified through a likelihood function $p(\mathbf{Y} | \beta)$. In the context of the regression model in Eq. (3.1), this likelihood would be considered conditional on the observed values of the predictors, i.e., the observed values of the predictors are considered as fixed. Finally, we need a prior distribution $p(\beta)$ for the model parameters β to incorporate our prior knowledge. Bayes’ rule then tells us that the posterior distribution $p(\beta | \mathbf{Y})$ is given by

$$p(\beta | \mathbf{Y}) \propto p(\beta) \times p(\mathbf{Y} | \beta). \quad (3.2)$$

The normalization constant can be calculated from the law of total probability if necessary. However, this calculation may not be always trivial so that simulation

methods, like MCMC, need to be employed. The posterior distribution is then used for further inference. For instance, we can look at its mean, mode, or other characteristics.

3.1.3 Linear Models

The linear model is one of the most popular forms for statistical modeling. Here, the functional relationship between the response and predictor is linear, i.e., $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$, where $\boldsymbol{\beta} \in \mathbb{R}^p$, and usually the assumption $\epsilon_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$ is made for the random errors. The linear model can be written in a matrix form for all cases $i \in \{1, \dots, n\}$ simultaneously as follows:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \quad (3.3)$$

where

$$\mathbf{Y} := \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} := \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} \quad \boldsymbol{\beta} := \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad \boldsymbol{\epsilon} := \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}. \quad (3.4)$$

The matrix \mathbf{X} is called the *design matrix*. Remember that each $\mathbf{x}_i \in \mathbb{R}^p$ is considered as a column vector, so \mathbf{X} is an $n \times p$ matrix.

3.1.4 Strong Duality and the Karush–Kuhn–Tucker Conditions

In this section, we briefly give the main duality result for nonlinear optimization that we will apply further. Assume we aim to minimize a function $f(\boldsymbol{\beta})$, where $\boldsymbol{\beta} \in B \subseteq \mathbb{R}^p$ subject to a constraint $h(\boldsymbol{\beta}) \leq 0$. In the following sections, we will have either $B = \mathbb{R}^p$ or $B = \mathbb{R}_+^p$ (i.e., the set of non-negative vectors in \mathbb{R}^p), although in principle B can be an arbitrary convex set. So, we try to find

$$f^* := \min_{\substack{\boldsymbol{\beta} \in B \\ h(\boldsymbol{\beta}) \leq 0}} f(\boldsymbol{\beta}). \quad (3.5)$$

One may think of the function $f(\cdot)$ as a least square criterion or a negative (log-) likelihood. Define now the *Lagrangian*:

$$\ell(\boldsymbol{\beta}, \lambda) := f(\boldsymbol{\beta}) + \lambda h(\boldsymbol{\beta}) \quad (3.6)$$

and the *Lagrange dual function*:

$$g(\lambda) := \min_{\beta \in B} \ell(\beta, \lambda). \quad (3.7)$$

Note that

$$\max_{\lambda \geq 0} g(\lambda) = \max_{\lambda \geq 0} \min_{\beta \in B} \ell(\beta, \lambda) \leq \max_{\lambda \geq 0} \min_{\substack{\beta \in B \\ h(\beta) \leq 0}} \ell(\beta, \lambda) \quad (3.8)$$

$$\leq \max_{\lambda \geq 0} \min_{\substack{\beta \in B \\ h(\beta) \leq 0}} f(\beta) = f^*. \quad (3.9)$$

This inequality holds in general. Strong duality tells us that, under certain conditions, the inequality becomes an equality [2, §5.2.3].

Theorem 3.1 (Strong Duality) *If f and h are convex functions, and $h(\beta) < 0$ for at least one $\beta \in B$, then*

$$\max_{\lambda \geq 0} g(\lambda) = \min_{\substack{\beta \in B \\ h(\beta) \leq 0}} f(\beta) = f^* \quad (3.10)$$

So, under strong duality, to minimize $f(\beta)$ over β subject to $h(\beta) \leq 0$, we can also instead maximize the Lagrange dual function over $\lambda \geq 0$. In that case, the *Karush–Kuhn–Tucker conditions* provide necessary and sufficient conditions for optimality.

Definition 3.1 (Subgradient) For any function F on B , we say that $v \in \mathbb{R}^p$ is a *subgradient* of F at β whenever

$$F(\beta') - F(\beta) \geq v^T (\beta' - \beta) \quad (3.11)$$

for all $\beta' \in B$. The set of all subgradients of F at β is denoted by $\partial F(\beta)$.

Theorem 3.2 (Karush–Kuhn–Tucker) *If f and h are convex functions, and $h(\beta) < 0$ for at least one $\beta \in B$, then $f(\beta) = f^*$ if*

$$\mathbf{0} \in \partial f(\beta) + \lambda \partial h(\beta) \quad (3.12)$$

$$\lambda h(\beta) = 0 \quad (3.13)$$

$$h(\beta) \leq 0 \quad (3.14)$$

$$\lambda \geq 0 \quad (3.15)$$

So, Eq. (3.12) is just a fancy way of writing that β is a global minimum of $f + \lambda h$, for a fixed value of λ . Equation (3.12) is called the *stationarity condition*. Equation (3.13) is called the *complementary slackness condition* and implies that either $\lambda = 0$ or $h(\beta) = 0$. The inequality $h(\beta) \leq 0$ is called *primal feasibility*, and the inequality $\lambda \geq 0$ is called *dual feasibility*.

To solve the Karush–Kuhn–Tucker conditions, we split the problem into two cases as per Eq. (3.13), $\lambda = 0$ and $h(\boldsymbol{\beta}) = 0$. We then solve Eq. (3.12) under each equality constraint. We throw away any solution that does not satisfy primal or dual feasibility and then choose the solution that achieves the lowest value.

For the case $\lambda = 0$, we need to find the global unconstrained minimum of f . If the primal feasibility constraint $h(\boldsymbol{\beta}) \leq 0$ is satisfied at the global minimum of f , then we have found a solution. Obviously, this solution must be the optimal solution of the original constrained problem as well.

If $h(\boldsymbol{\beta}) > 0$ at the global minimum of f , then we need to find the minimum of f under the constraint that $h(\boldsymbol{\beta}) = 0$. We could do so by finding a joint solution to the system of equations formed by Eq. (3.12) and $h(\boldsymbol{\beta}) = 0$. Alternatively, we could gradually increase λ until the global unconstrained minimum $g(\lambda)$ of $f + \lambda h$ satisfies $h(\boldsymbol{\beta}) = 0$. Indeed, due to the form of the objective function, increasing λ will favor $\boldsymbol{\beta}$ that have lower values for $h(\boldsymbol{\beta})$, so eventually, $h(\boldsymbol{\beta}) = 0$. By strong duality, we also know that finding this λ is equivalent to maximizing the Lagrange dual function $g(\lambda)$ over $\lambda \geq 0$.

3.2 Parameter Estimation

In a statistical modeling problem our task is to estimate $\boldsymbol{\beta}$ from the data \mathbf{Y} and \mathbf{X} . There are several methods to estimate these parameters in a linear model. We will discuss some of them and their properties.

3.2.1 Ordinary Least Squares

In OLS [5], we estimate the parameters by minimizing the sum of the squared errors:

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} := \arg \min_{\boldsymbol{\beta}} R(\boldsymbol{\beta}) \quad (3.16)$$

where

$$R(\boldsymbol{\beta}) := \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \quad (3.17)$$

We have used $\|\cdot\|_2$ to denote the standard Euclidean norm that is $\|\mathbf{z}\|_2 := \sqrt{\sum_{i=1}^n z_i^2}$. A necessary condition to have a minimum for Eq. (3.17) is

$$\frac{\partial}{\partial \boldsymbol{\beta}} R(\boldsymbol{\beta}) = -2\mathbf{X}^T \mathbf{Y} + 2(\mathbf{X}^T \mathbf{X})\boldsymbol{\beta} = 0. \quad (3.18)$$

Therefore, if $\mathbf{X}^T \mathbf{X}$ is invertible (this requires that the number of observations, n , is larger or equal than the total number of predictors, p), then the OLS estimator is given by

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}, \quad (3.19)$$

where $(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is the Moore–Penrose inverse of \mathbf{X} .

The *Gauss–Markov theorem* states that when the errors are uncorrelated with expectation zero and constant variance, then the OLS estimate is the best linear unbiased estimator.

Two issues that often arise are:

1. If $p > n$ then $\mathbf{X}^T \mathbf{X}$ is singular; hence Eq. (3.18) has no unique solution.
2. Even if $p \leq n$, p may still be much larger than needed, and we may wish to identify sparse solutions where unnecessary parameters are set to zero. In other words, we may wish to perform variable selection as part of our statistical inference.

3.2.2 Non-Negative Garrote

The non-negative garrote was introduced by Breiman [3]. It is a two-stage procedure that gives a sparse solution. It has a close relationship to the LASSO; however as a starting point of the problem, the OLS estimates are needed. Given the initial estimate $\hat{\boldsymbol{\beta}}^{\text{OLS}} \in \mathbb{R}^p$, we solve the following optimization problem over $\mathbf{c} = (c_1, c_2, \dots, c_p)^T$:

$$\hat{\mathbf{c}} = \arg \min_{\substack{\mathbf{c} \geq 0 \\ \|\mathbf{c}\|_1 \leq t}} \|\mathbf{Y} - \mathbf{X} \mathbf{C} \hat{\boldsymbol{\beta}}^{\text{OLS}}\|_2^2 \quad (3.20)$$

where $\mathbf{C} := \text{diag}(\mathbf{c}) \in \mathbb{R}^{p \times p}$, and where $\|\cdot\|_1$ denotes the l_1 -norm; that is $\|\mathbf{c}\|_1 = \sum_{i=1}^p |c_i|$. We get the final non-negative garrote parameter estimate $\hat{\boldsymbol{\beta}}$ by setting $\hat{\boldsymbol{\beta}}_i = \hat{c}_i \hat{\boldsymbol{\beta}}_i^{\text{OLS}}$ for each $i \in \{1, 2, \dots, p\}$.

Equivalently, we can solve the dual problem, by introducing a Lagrangian multiplier λ for the constraint $\|\mathbf{c}\|_1 - t \leq 0$ [16], similar to what we discussed in Sect. 3.1.4:

$$\max_{\lambda \geq 0} \min_{\mathbf{c} \geq 0} \left(\|\mathbf{Y} - \mathbf{X} \mathbf{C} \hat{\boldsymbol{\beta}}^{\text{OLS}}\|_2^2 + \lambda (\|\mathbf{c}\|_1 - t) \right) \quad (3.21)$$

Effectively, we thus need to solve

$$\hat{c}_\lambda = \arg \min_{c \geq 0} \left(\|Y - XC\hat{\beta}^{\text{OLS}}\|_2^2 + \lambda \|c\|_1 \right) \quad (3.22)$$

where the Lagrange multiplier $\lambda \geq 0$ can be interpreted as a regularization weight. If $\|\hat{c}_\lambda\|_1 \leq t$ for $\lambda = 0$, then we are done. Otherwise, λ is calibrated until $\|\hat{c}_\lambda\|_1 = t$, as we discussed in Sect. 3.1.4. This value for λ is also the value that achieves the maximum in Eq. (3.21). If the columns of the design matrix X are orthogonal (i.e., $X^T X = I$), then the explicit solution of Eq. (3.22) is given by [26]

$$\hat{c}_{\lambda i} = \max \left\{ 0, 1 - \frac{\lambda}{(\hat{\beta}_i^{\text{OLS}})^2} \right\}. \quad (3.23)$$

Consequently, in this case, if the coefficient $\hat{\beta}_i^{\text{OLS}}$ of a predictor is less than $\sqrt{\lambda}$, then $\hat{c}_{\lambda i} = 0$, and therefore also $\hat{\beta}_i = \hat{c}_{\lambda i} \hat{\beta}_i^{\text{OLS}} = 0$. In this way, larger λ will produce sparser solutions.

The starting point of this method depends on the least square estimates $\hat{\beta}^{\text{OLS}}$. Therefore, if $p > n$, then no unique solution is available. However, alternative initial estimators, such as the LASSO, can be used in this case [26].

3.2.3 Regularization Under l_q Penalty

Unfortunately, the non-negative garrote in Eq. (3.20) still fails to deliver when we have no least square estimate to start from, which happens, for instance, when we have more predictors than observations. To solve this, we can use a different method, where no initial estimate is needed. The basic idea is to add a penalty term to the least square problem, in order to penalize non-zero parameter values. This can be done in the following way:

$$\hat{\beta}_\lambda = \arg \min_{\beta} \left(\frac{1}{2} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_q^q \right) \quad (3.24)$$

where $q \geq 0$ determines the shape of the penalty, and $\lambda \geq 0$ determines the strength of the penalty. Here,

$$\|z\|_q^q := \begin{cases} \sum_{i=1}^n |z_i|^q & \text{if } q > 0 \\ \sum_{i=1}^n I_{z_i \neq 0} & \text{if } q = 0 \end{cases} \quad (3.25)$$

where $I_{z_i \neq 0} = 1$ if $z_i \neq 0$ and 0 otherwise. So, $\|z\|_0^0$ simply counts the number of non-zero components of z .

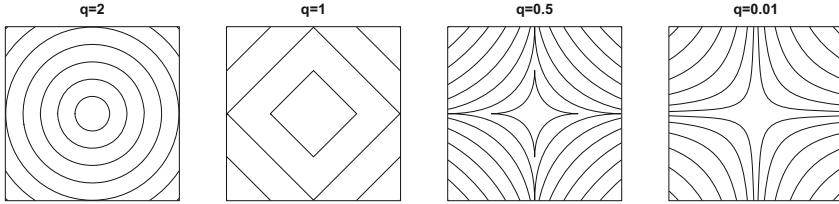


Fig. 3.2 Contour plots of different l_q penalty functions

For different values of q we have different types of regularization. This leads to ridge regression for $q = 2$, LASSO for $q = 1$, and best subset selection method for $q = 0$ [16].

In Fig. 3.2, we illustrate some contour plots of the l_q penalty function, for different values of q . As will be illustrated in Sect. 3.3.1, it is the “spiked” shape of the contours which leads to sparsity; in other words all penalties with $q \leq 1$ will lead to sparse estimators. However, for $q < 1$, the l_q penalty function is no longer convex, as can be seen from the contour plots. Therefore, $q = 1$ is the only value for which the problem is convex and allows sparse solutions.

3.3 The LASSO

The LASSO estimator was first proposed by Tibshirani [24]. The objective is to solve the OLS problem but subject to an additional constraint on the 1-norm of the parameters, as follows:

$$\min_{\beta: \|\beta\|_1 \leq t} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 \right). \tag{3.26}$$

It is usually assumed that \mathbf{X} and \mathbf{Y} are standardized to mean 0. Otherwise, they can always be standardized without any loss of generality.

3.3.1 Solving the LASSO Optimization Problem

By strong duality (see Theorem 3.1 in Sect. 3.1.4), equivalently, we can solve the dual problem, by introducing a Lagrangian multiplier λ for the constraint $\|\beta\|_1 - t \leq 0$:

$$\max_{\lambda \geq 0} \min_{\beta} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda(\|\beta\|_1 - t) \right). \tag{3.27}$$

For the inner minimization problem, we need to find

$$\hat{\boldsymbol{\beta}}_\lambda := \arg \min_{\boldsymbol{\beta}} \left(\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right). \quad (3.28)$$

From the discussion in Sect. 3.1.4, we know that if $\|\hat{\boldsymbol{\beta}}_0\|_1 \leq t$, then the solution is immediately given by $\hat{\boldsymbol{\beta}}_0$ (note that $\hat{\boldsymbol{\beta}}_0 = \hat{\boldsymbol{\beta}}^{\text{OLS}}$). If $\|\hat{\boldsymbol{\beta}}_0\|_1 > t$, then we need find that value for $\lambda \geq 0$ for which $\|\hat{\boldsymbol{\beta}}_\lambda\|_1 = t$, and the solution is then given by the corresponding $\hat{\boldsymbol{\beta}}_\lambda$. In either case, this λ is also the λ which achieves the maximum in Eq. (3.27) and which solves the Karush–Kuhn–Tucker conditions in theorem 3.2.

Let us derive the stationarity condition (Eq. (3.12) in Sect. 3.1.4) of the Karush–Kuhn–Tucker equations, specifically for the LASSO. As we saw, along with complementary slackness (either $\lambda = 0$ or $\|\boldsymbol{\beta}\|_1 = t$) and feasibility ($\lambda \geq 0$ and $\|\boldsymbol{\beta}\|_1 \leq t$), this condition fully characterizes the optimality of our solution.

For the LASSO, the Lagrangian is given by

$$\frac{1}{2} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda(\|\boldsymbol{\beta}\|_1 - t).$$

The stationarity condition says that the subgradient with respect to $\boldsymbol{\beta}$ of this Lagrangian must contain the origin, i.e., we need that

$$\mathbf{0} \in -\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \partial \|\boldsymbol{\beta}\|_1. \quad (3.29)$$

It can be shown that [20, §3.1.5]

$$\partial \|\boldsymbol{\beta}\|_1 = \text{sign}(\beta_1) \times \cdots \times \text{sign}(\beta_p) \quad (3.30)$$

where

$$\text{sign}(\beta_j) := \begin{cases} \{-1\} & \text{if } \beta_j < 0 \\ [-1, 1] & \text{if } \beta_j = 0 \\ \{1\} & \text{if } \beta_j > 0. \end{cases} \quad (3.31)$$

Therefore, we can write Eq. (3.29) in the following way

$$\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) = \lambda \mathbf{s} \quad (3.32)$$

where $\mathbf{s} = (s_1, s_2, \dots, s_p)$ are auxiliary variables subject to the constraint $s_j \in \text{sign}(\beta_j)$.

When the columns of \mathbf{X} are orthogonal (this holds, for instance, when there is only one predictor) and are standardized such that $\mathbf{X}^T \mathbf{X} = \mathbf{I}$, the solution to this system can be expressed as a thresholded version of the OLS [16]:

Fig. 3.3 Soft-thresholding function $S_\lambda(x)$ for $\lambda = 1$

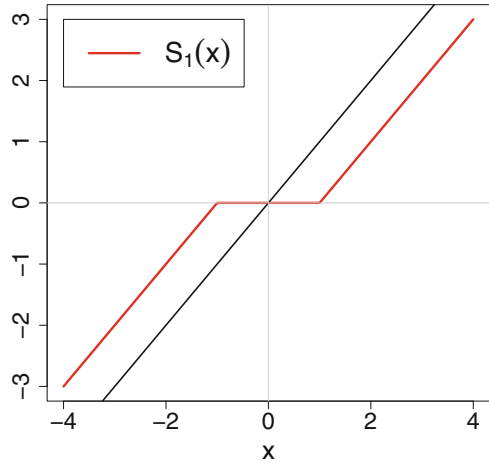
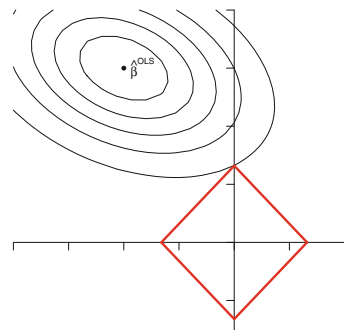


Fig. 3.4 Relationship between the OLS estimate and the l_1 constraint imposed by the LASSO (red), adapted from [15]



$$\hat{\beta}_{\lambda j} = S_\lambda(\hat{\beta}_j^{\text{OLS}}) \tag{3.33}$$

with *soft-thresholding operator* (see Fig. 3.3)

$$S_\lambda(\beta_j) := \text{sign}(\beta_j) \max\{0, |\beta_j| - \lambda\}. \tag{3.34}$$

Otherwise, the solution can still be expressed through an iterative execution of soft-thresholding operations [16].

The contour lines in Fig. 3.4 illustrate why and how the LASSO works. The contours refer to the OLS problem, and the diamond corresponds to the constraint $\|\beta\|_1 = t$. Remember that $\hat{\beta}^{\text{OLS}} = \hat{\beta}_0$, so the figure depicts the case where $\|\hat{\beta}_0\| > t$. We want the point on the diamond closest to the OLS. This is likely to lie on the axes, hence setting smaller parameters to 0.

3.3.2 Cross-Validation

Cross-validation is a commonly used method to identify the optimal value of a tuning parameter, which is in our case the penalty parameter λ . It is based on minimizing an estimate of the prediction error. In cross-validation, we use one part of the data to fit the LASSO model and the other part of the data to validate it [15].

We fix initially a dense grid of values of λ , i.e., λ is discretized with small step-sizes over a suitable range which reflects the scope of the regularization trade-off that we are willing to consider. The dataset is then divided into K equally sized partitions. We assume for simplicity that K is a divisor of n so that each partition contains n/K elements. For each fixed value of λ of the grid, and the k 'th partition, $k = 1, \dots, K$, we fit the LASSO model using the remaining $K - 1$ parts and calculate the prediction error of the fitted model. Specifically, denote $\hat{\beta}_\lambda^{-k}$ the parameter vector obtained under a penalty of λ when omitting the k 'th partition, so that $\mathbf{x}_i^T \hat{\beta}_\lambda^{-k}$ is the corresponding fitted model under predictor \mathbf{x}_i . Then the prediction error for the k 'th partition is

$$P_k(\lambda) = \frac{K}{n} \sum_{i=1}^{n/K} L(y_i, \mathbf{x}_i^T \hat{\beta}_\lambda^{-k}) \quad (3.35)$$

where, for the linear model (Eq. (3.3)), the loss function L is just the squared error. We repeat this step for every $k = 1, 2, \dots, K$ and combine the values of $P_k(\lambda)$ to find the average prediction error, $P(\lambda) = K^{-1} \sum_{k=1}^K P_k(\lambda)$. This is then repeated for every value of λ in the grid, and we choose the value of λ which minimizes $P(\lambda)$ [16].

For smaller values of λ , the LASSO estimators contain more predictors which may lead to an over-fitted model. However, for larger values of λ , the model has fewer predictors leading to sparsity and producing a more easily interpretable model.

To avoid misunderstandings, it is noted that the problem of finding the optimal λ (in the sense of minimal prediction error), as discussed in this subsection, is very different from, and entirely unrelated to, the problem of maximizing over λ as, for instance, in Eq. (3.27). The latter is a purely formal operation which ensures mathematical equivalence of the two dual versions of the LASSO optimization problem and does not imply any statement on the best choice of λ .

3.3.2.1 Example: Gaia Dataset

Figure 3.5 represents the cross-validation curve for the Gaia dataset. Here we have taken normalized data to get rid of scalability. The graph is consistent with the property of cross-validation, i.e., we can see that for smaller values of λ the number of predictors is higher and for larger values of λ the number of predictors gets

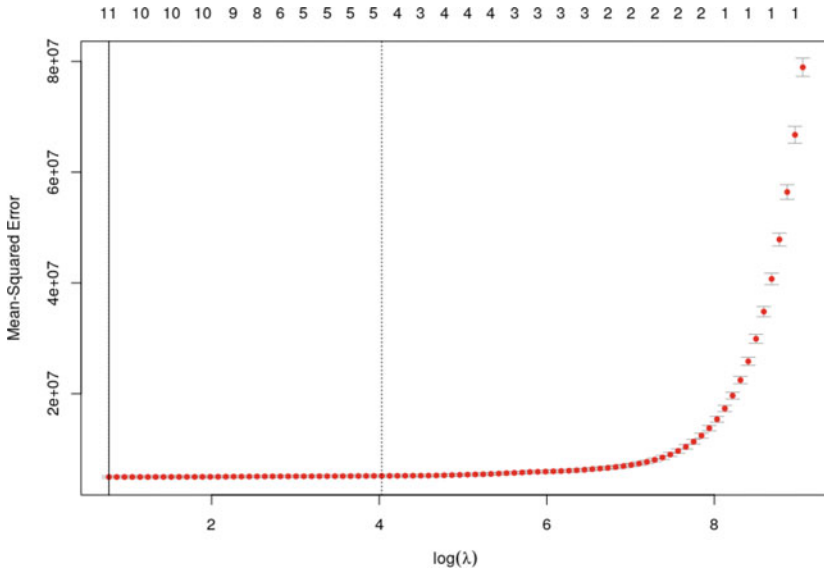


Fig. 3.5 Cross-validation curve for the Gaia dataset, with the number of selected predictor variables as a function of $\log(\lambda)$ given on top of the plot

reduced. Here, $\log(\lambda)$ is used as the tuning parameter, increased values of which lead to reduced numbers of included variables (note that \log denotes the natural logarithm throughout this chapter). From the cross-validation curve, we get the value of $\log(\lambda)$ to be approximately 0.775 (shown by the solid vertical line), and hence the prediction error of the LASSO-fitted model is minimal at $\lambda \approx 2.17$. We use this value to estimate the coefficients of the parameters. Note that the plot for this dataset is somewhat unusual, as the minimum falls close to the boundary (solid vertical line); compare further with Fig. 3.11 for a more typical appearance.

Figure 3.6 shows the coefficient path of the parameters, i.e., the change in coefficients of the predictors as a function of λ . The black vertical line denotes the value of $\log(\lambda)$ for which the prediction error is minimal. For this particular value of λ , we see that there are only 11 non-zero parameters, and others are shrunk towards zero.

For the cross-validation method for LASSO, we have used the **glmnet** [11] package in R. It is noted at this occasion that this software by default also draws a second vertical line in the cross-validation plot (which is dotted in Fig. 3.5), which indicates the largest value of $\log(\lambda)$ which is less than one standard error (calculated for each λ from the $P_k(\lambda)$, $k = 1, \dots, K$) away from the minimum [16]. Arguably this gives an even sparser solution which is statistically not distinguishable from the one obtained under the minimum. We do not follow this line of reasoning in this exposition and work with the estimator under the “optimal” λ at all occasions.

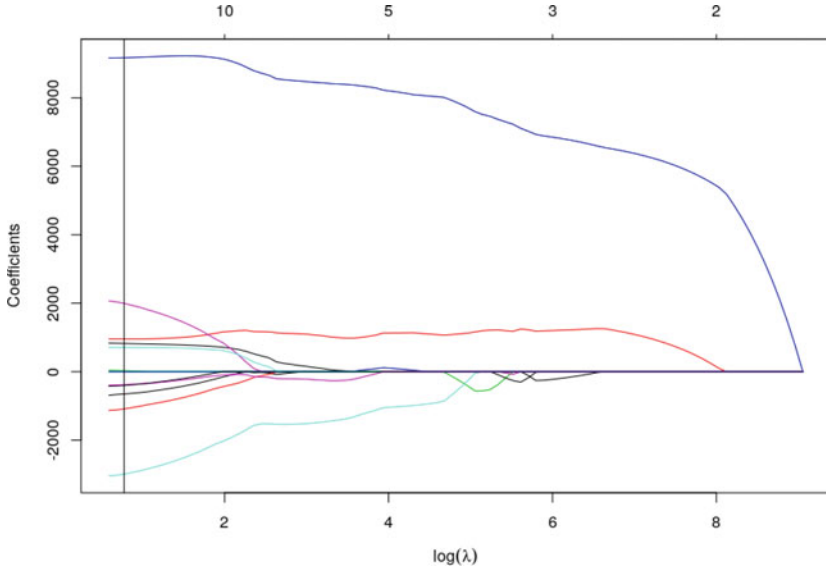


Fig. 3.6 Coefficient path of the parameters for the Gaia dataset

3.4 Uncertainty Quantification

3.4.1 Refit-LASSO

The “refit”-LASSO is one of the possible ways to quantify system uncertainty of a LASSO-fitted model. The simple idea is to use the “important” (non-zero) variables selected by the LASSO procedure in a subsequent OLS fit.

We implement a slight modification of this idea. We carry out the entire cross-validation procedure multiple times with random partitions, which gives us different optimized λ for each run, producing an ensemble of possible estimates of β . We then let the ensemble vote on the inclusion of the variables into the model. We will consider variables as important, if they have not been shrunk to 0 for a pre-defined proportion of the runs. Then we apply an OLS fit on the important variables to get the refit-LASSO estimates. Standard errors of the j 'th parameter estimate, $\hat{\beta}_j$, are then obtained as $s\sqrt{(X^T X)_j^{-1}}$, where the suffix j indicates the j 'th diagonal element taken after application of the inverse, and s^2 denotes the unbiased estimator of σ^2 .

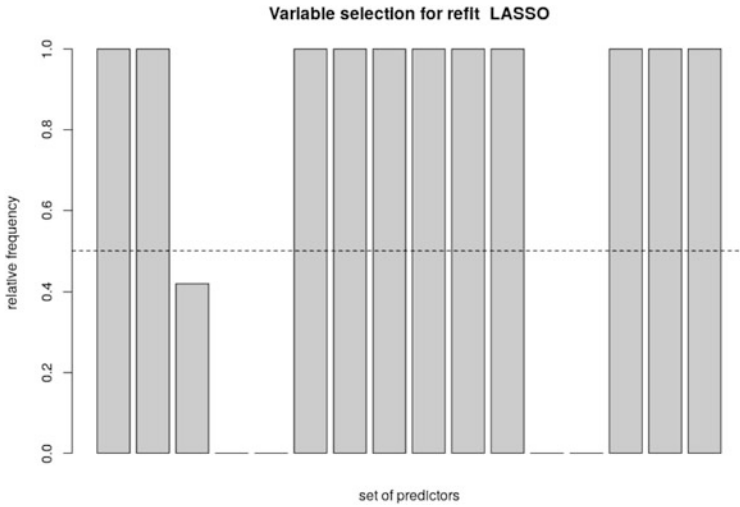


Fig. 3.7 Relative frequency of occurrence of variables, for refit-LASSO applied on the Gaia dataset

3.4.1.1 Example: Gaia Dataset

We applied the refit-LASSO on the Gaia dataset. We have taken 100 simulation runs for the selection of important variables. The result is displayed in Fig. 3.7. We set the desired proportion of inclusion at 50% as indicated by a horizontal line. Then we have applied OLS fit on the important variables; in Table 3.1 we show the standard error of our prediction with its “t-value” and corresponding probability. We also give a comparison between the refit-LASSO estimates and the original cross-validated LASSO estimates in the last two columns.

We notice from the Fig. 3.7 that the third variable appeared to be important in several runs. However, it is not important in most of the runs.

3.4.2 Bootstrap Method

Bootstrap is a general frequentist method to quantify statistical accuracy, where one randomly draws samples from a given training dataset with replacement, the sample size being equal to that of the original training dataset. This is done for B times (often multiples of 1000). Then one fits the model to each of these B datasets and examines the empirical distributions of the estimated parameters.

Table 3.1 Summary of refit-LASSO for the Gaia dataset. The column “Estimate” gives the parameter estimates from the refitted model using the selected variables. “Original” estimates refer to a (single) initial cross-validated LASSO execution as discussed in Sect. 3.3.2, and “Difference” refers to the difference between the refit-LASSO and original estimates

Predictors	Estimate	Std. error	t value	$Pr(> t)$	Original	Difference
band1	841.04	140.89	5.97	0.00	823.53	17.51
band2	1001.36	298.78	3.35	0.00	954.10	47.26
band6	8960.42	434.64	20.62	0.00	9169.52	-209.09
band7	-3664.57	257.19	-14.25	0.00	-2992.80	-671.77
band8	2842.23	260.48	10.91	0.00	1995.79	846.44
band9	-987.10	201.13	-4.91	0.00	-651.95	-335.15
band10	-1584.91	213.89	-7.41	0.00	-1088.03	-496.88
band11	150.19	175.58	0.86	0.39	28.85	121.33
band14	685.64	204.44	3.35	0.00	708.89	-23.25
band15	-588.20	234.04	-2.51	0.01	-381.77	-206.43
band16	-641.26	259.41	-2.47	0.01	-401.16	-240.10

3.4.2.1 Bootstrap for LASSO

For the LASSO estimation methodology as outlined in Sects. 3.3.1 and 3.3.2, the bootstrap technique is applied straightforwardly, but it has to be ensured that the selection of λ through cross-validation is part of the uncertainty being assessed. Specifically, for each sample dataset obtained through the aforementioned bootstrap routine, we perform cross-validation to obtain the minimal prediction error. This gives us a selected value of λ and hence a parameter estimate $\hat{\beta}_\lambda$ for each bootstrap sample. Then, we use these to calculate the bootstrap standard deviations or empirical distributions of the parameters.

3.4.2.2 Example: Gaia Dataset

At first, we get a one-time LASSO estimate using the cross-validation method. Then we take 1000 bootstrap replicates of the original Gaia dataset to calculate the bootstrap statistics. In Table 3.2 we display the summary of our bootstrap result. In addition to the bootstrap mean, median, and standard deviation, we also calculated the bootstrap bias using the formula

$$\text{Bias} = \text{Initial Estimate} - \text{Bootstrap Mean}$$

In Fig. 3.8, we visualize the bootstrapped distribution of the parameters through box-plots.

Clearly, it can be seen from Table 3.2 and Fig. 3.8 that band3, band4, band5, band12, and band13 are the non-important parameters. While the mean for band3 and band13 is not very close to 0, they still act as non-important parameters with median being 0.

Table 3.2 Summary of bootstrap estimates for the Gaia dataset. The lower and upper bounds of the 95% confidence intervals for model parameters are obtained as the 2.5% and 97.5% quantiles of the empirical bootstrap distributions

Predictors	Mean	Median	Bias	SD	CI-lower	CI-upper
band1	827.98	835.75	9.33	132.04	483.41	1062.81
band2	991.57	986.84	37.78	333.28	350.15	1655.07
band3	10.21	0.00	10.21	64.39	-30.62	182.39
band4	-2.21	0.00	-2.21	27.26	-67.58	46.23
band5	0.25	0.00	0.25	2.50	0.00	0.87
band6	9127.26	9137.34	-49.87	366.25	8421.19	9797.16
band7	-2984.39	-3019.35	-37.65	338.35	-3441.90	-1557.63
band8	1998.24	2059.49	58.57	429.29	0.84	2486.86
band9	-678.97	-692.91	-46.31	188.08	-1013.58	-78.25
band10	-1092.59	-1123.82	-42.44	226.78	-1392.43	-127.41
band11	58.00	21.52	37.91	78.82	-3.37	254.11
band12	-6.95	0.00	-6.95	22.94	-80.57	0.24
band13	22.90	0.00	22.90	32.18	-0.39	102.34
band14	680.22	697.12	-27.19	147.25	215.06	920.35
band15	-400.80	-405.13	-30.81	127.43	-637.49	-139.13
band16	-391.75	-398.42	-8.78	136.66	-638.43	0.00

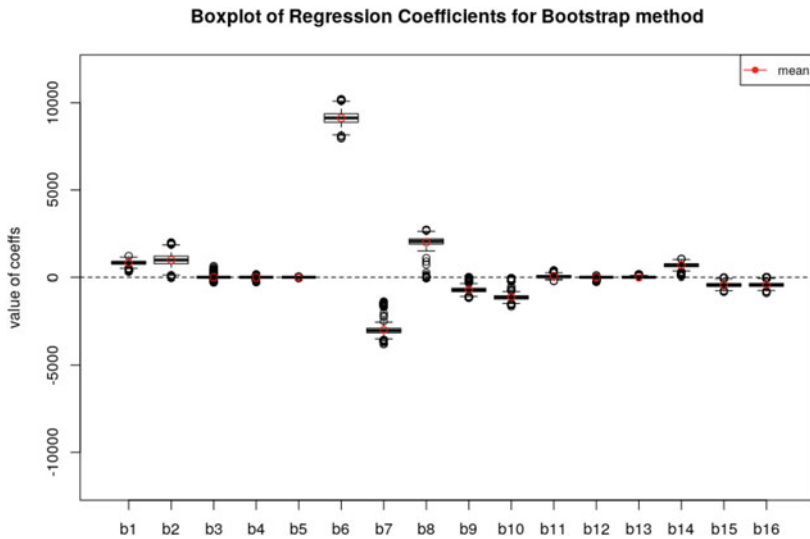


Fig. 3.8 Bootstrapped distribution of the parameters in the Gaia dataset

3.4.3 Bayesian LASSO

The Bayesian methodology provides a natural way to quantify the model uncertainty in a LASSO-fitted model. To motivate this approach, recall firstly that, under the assumption $\boldsymbol{\epsilon} \sim N(0, \sigma^2 \mathbf{I})$, we can write the likelihood of model (3.3) in the following way,

$$\begin{aligned} p(\mathbf{Y} | \mathbf{X}, \boldsymbol{\beta}) &\propto e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n \epsilon_i^2} \\ &\propto e^{-\frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}. \end{aligned} \quad (3.36)$$

Tibshirani [24] suggested using a Laplace prior

$$p(\boldsymbol{\beta}) \propto e^{-\lambda \|\boldsymbol{\beta}\|_1} \quad (3.37)$$

for the model parameters, yielding the following posterior,

$$\begin{aligned} p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{Y}) &\propto p(\mathbf{Y} | \mathbf{X}, \boldsymbol{\beta}) \times p(\boldsymbol{\beta}) \\ &\propto e^{-\left(\frac{1}{2\sigma^2} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1\right)} \end{aligned} \quad (3.38)$$

It is a well-established result that the mode of (3.38), i.e., the posterior mode of $\boldsymbol{\beta}$ under Laplace priors, corresponds just to the frequentist LASSO estimate [18, 22, 24]. Draws from this posterior are not necessarily sparse but still can be used to assess uncertainty of model parameters [16].

The Bayesian LASSO has been implemented in several different facets, which differ essentially in the way that sparsity is induced and in the way that the regularization parameter is handled. In 2008, Park and Casella [22] proposed a hierarchical mixture model for parameter estimation:

$$\begin{aligned} \mathbf{Y} | \mu, \mathbf{X}\boldsymbol{\beta}, \sigma^2 &\sim N_n(\mu \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n), \\ \boldsymbol{\beta} | \sigma^2, \tau_1^2, \dots, \tau_p^2 &\sim N_p(\mathbf{0}_p, \sigma^2 \mathbf{D}_\tau) \\ \mathbf{D}_\tau &= \text{diag}(\tau_1^2, \dots, \tau_p^2), \\ \sigma^2, \tau_1^2, \dots, \tau_p^2 &\sim \pi(\sigma^2) d\sigma^2 \prod_{j=1}^p \frac{\lambda^2}{2} e^{-\lambda^2 \tau_j^2 / 2} d\tau_j^2, \\ \sigma^2, \tau_1^2, \dots, \tau_p^2 &> 0. \end{aligned} \quad (3.39)$$

After marginalizing over $\tau_1^2, \dots, \tau_p^2$, we get the conditional prior on $\boldsymbol{\beta}$ of the following form

$$\pi(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^p \frac{\lambda}{2\sigma} e^{-\lambda|\beta_j|\sigma}. \quad (3.40)$$

For the choice of the LASSO penalty parameter, Park and Casella suggested two different techniques. Firstly, they suggested the possibility of using marginal maximum likelihood estimates for the choice of λ . They considered a Monte Carlo EM algorithm which, in iteration k , updates the parameter λ using the iterative scheme

$$\lambda_k = \sqrt{\frac{2p}{\sum_{j=1}^p E_{\lambda_{k-1}}[\tau_j^2|\mathbf{Y}]}}}, \quad (3.41)$$

where \mathbf{Y} is assumed to be centered, and the conditional expectation is estimated via averages of a Gibbs sample. For $p < n$, the initial value λ_0 was suggested to be

$$\lambda_0 = \frac{p\sqrt{\hat{\sigma}_{\text{OLS}}^2}}{\sum_{j=1}^p |\hat{\beta}_j^{\text{OLS}}|},$$

where $\hat{\sigma}_{\text{OLS}}^2$ and $\hat{\beta}_j^{\text{OLS}}$ are OLS estimates. In another approach, they discussed the possibility of using gamma priors on λ^2 :

$$\pi(\lambda^2) = \frac{\delta^r}{\Gamma(r)} (\lambda^2)^{r-1} e^{-\delta\lambda^2}; \quad \lambda^2 > 0 \ (r > 0, \delta > 0), \quad (3.42)$$

where r is the shape parameter and δ the rate parameter. Lykou and Ntzoufras [18] used gamma priors for λ and developed a concept for specification of the hyperparameters based on Bayes factors which evaluate the evidence for inclusion of the respective predictor variables.

3.4.3.1 Example: Gaia Dataset

We obtained the posterior distribution of the parameters for the Gaia dataset using the `blasso` function from the `monomvn` [13] package in R. For the choice of the LASSO penalty parameter λ , we used marginal maximum likelihood estimates, as mentioned earlier. We drew 1000 posterior samples from this distribution, which are displayed in Fig. 3.9.

It can be seen that the output from the Bayesian method is similar to that of the Bootstrap method. For a better comparison between the methods, we also show the standard errors for the coefficient estimates of each important variable in Fig. 3.10.

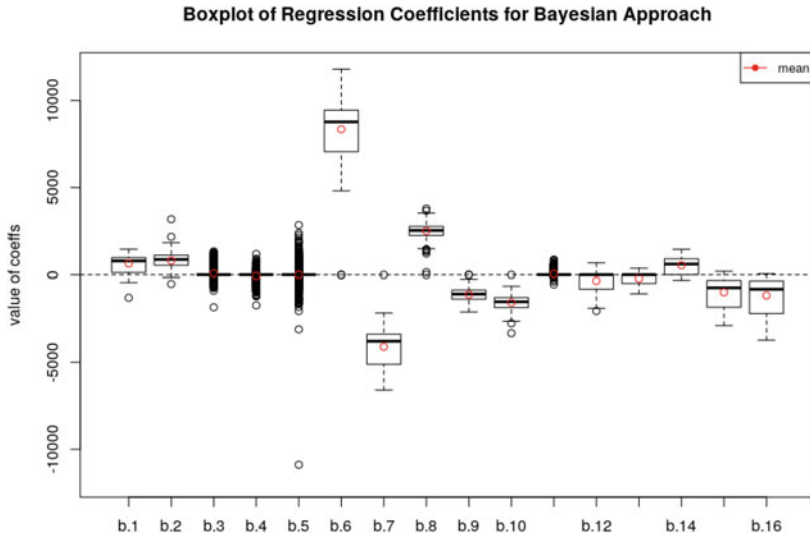


Fig. 3.9 Posterior distribution of the parameters in the Gaia dataset

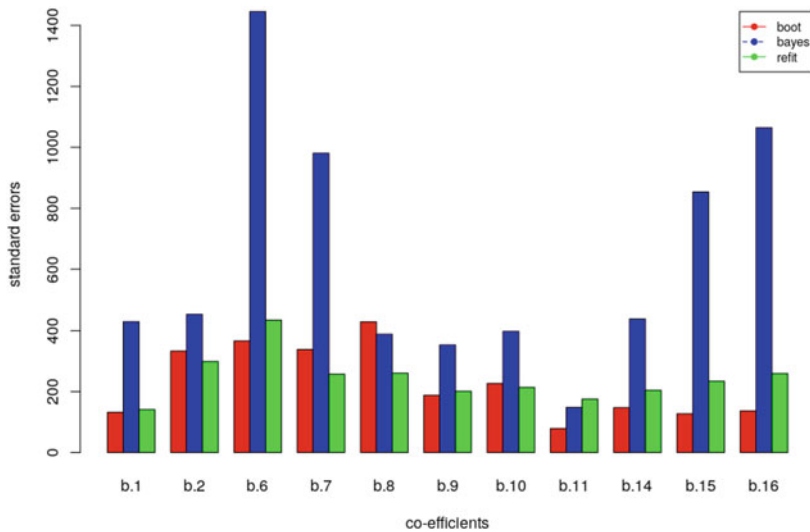


Fig. 3.10 Standard errors of LASSO-based parameter estimates for the Gaia dataset, obtained from different methods

3.5 LASSO for Classification

Recall the linear model in a row-wise notation, $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i$, or $E(y_i | \mathbf{x}_i) = \mathbf{x}_i^T \boldsymbol{\beta}$, which makes the implicit assumption on the distribution of the response variable:

$$y_i \sim N(\mathbf{x}_i^T \boldsymbol{\beta}, \sigma^2).$$

However, this assumption is too restrictive for many real data situations.

One can use generalized linear models to relax the assumption of normality. We introduce a function g , which acts as a link function such that

$$g(E(y_i | \mathbf{x}_i)) = \mathbf{x}_i^T \boldsymbol{\beta}; \quad (3.43)$$

here, y_i can possess any exponential family distribution, such as Poisson, Binomial, or Gamma. Note that if $y_i \in \{0, 1\}$ then

$$\mu_i \equiv E(y_i | \mathbf{x}_i) = P(y_i = 1 | \mathbf{x}_i); \quad (3.44)$$

hence we can (for our purposes) define

Definition 3.2 (Classification) Classification is the process of carrying out a regression problem with 0/1-valued response and allocating observations to one of the two classes according to the decision rule $\mu_i \geq 0.5$.

3.5.1 Logistic Regression

In logistic regression we start with the logistic model,

$$\log \frac{\mu_i}{1 - \mu_i} = \mathbf{x}_i^T \boldsymbol{\beta} \quad (3.45)$$

with “logit” link function $g(\mu_i) = \log \frac{\mu_i}{1 - \mu_i}$. An alternative formulation of Eq. (3.45) is

$$P(y_i = 1 | \mathbf{x}_i) = h(\mathbf{x}_i^T \boldsymbol{\beta}) \quad (3.46)$$

where the *logistic function*

$$h(t) = \frac{\exp(t)}{1 + \exp(t)} \quad (3.47)$$

maps the range $(-\infty, \infty) \rightarrow [-1, 1]$. The parameters in the logistic model are estimated through an iteratively weighted least squares technique known as “Fisher Scoring,” for details of which we refer to [9].

Example 3.2 (Sonar Dataset) Gorman and Sejnowski used this dataset in their study of the classification of sonar signals using a neural network [12]. The objective of the study was to discriminate between sonar signals bounced off a metal cylinder and a cylindrical rock. Each observation is a set of 60 numbers

(serving as predictor variables) in the range 0.0–1.0. Each number represents the energy within a particular frequency band, integrated over a certain period of time. The label associated with each response contains the letter “R” if the object is a rock and “M” if it is a mine (metal cylinder). There are total of 208 observations in this dataset [17]. Here, due to computational limitations, we have taken the first 48 predictors of the Sonar dataset and used the standardized form to handle numerical scaling issues, throughout the examples.

3.5.1.1 Cross-Validation

We apply cross-validation onto the Sonar dataset and investigate the achieved sparsity as compared with the original model with 48 different predictors. The result of the cross-validation procedure is displayed in Fig. 3.11. The prediction error for this purpose is calculated as in Eq. (3.35), but now the loss function L is given by the deviance (i.e., two times the difference of saturated and model log likelihood [9]). From Fig. 3.11 we find that the prediction error is minimal when $\log \lambda = -3.672$, so $\lambda = 0.0254$. Using this value of λ , we calculate the coefficients of the parameters. For this particular dataset, LASSO eliminates 29 predictors and reduces the number of retained variables to 19. In Fig. 3.12, we illustrate the coefficient path of the parameters.

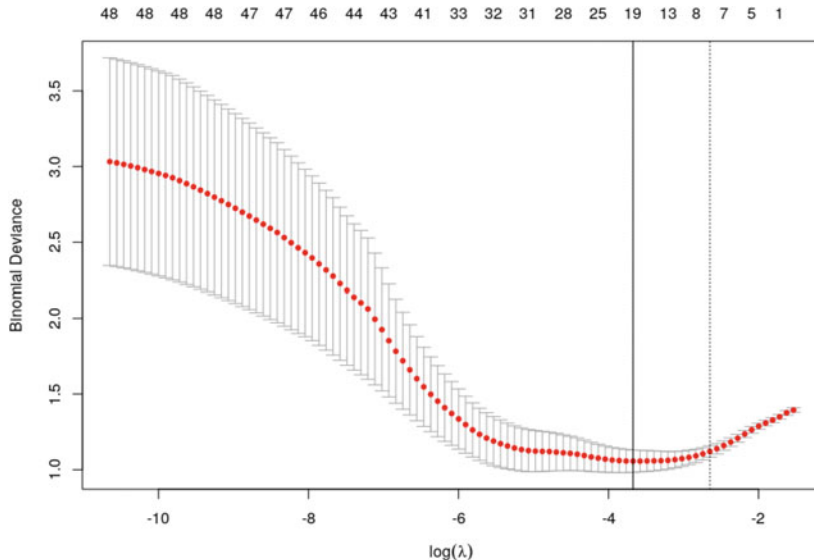


Fig. 3.11 Cross-validation curve for Sonar dataset

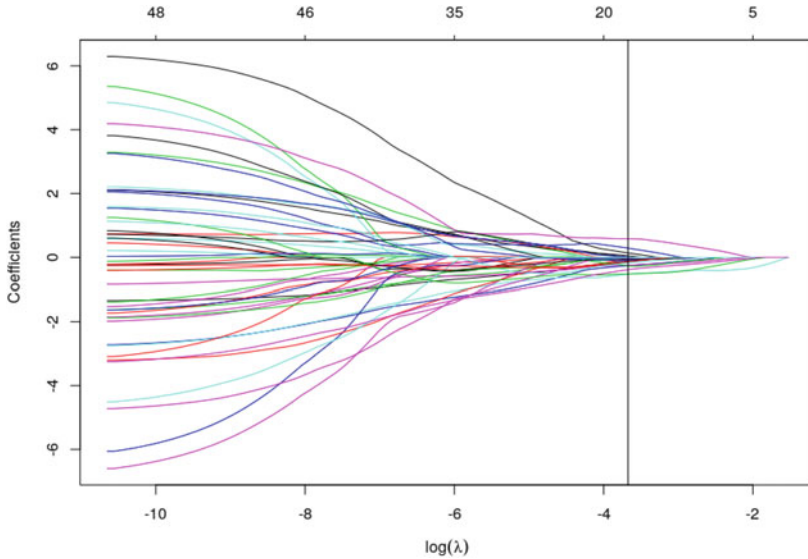


Fig. 3.12 Coefficient path of the parameters for the Sonar dataset

3.5.2 *Uncertainty Quantification*

Here, we will discuss uncertainty quantification for the LASSO under the logistic model, by way of application on the Sonar dataset.

3.5.2.1 **Refit-LASSO**

We applied the refit-LASSO method on the Sonar dataset. We carried out 100 cross-validation runs with randomized partitions to check the behavior of variable selection. We considered variables as important if they appeared to be non-zero in 50 or more runs. We illustrate the selection of important variable in Fig. 3.13. Then we applied logistic regression on the important variables. We used the `glm` package in R for model fitting. The corresponding refit-LASSO estimates are given in Table 3.3.

3.5.2.2 **Bootstrap**

We applied the bootstrap method on the Sonar dataset with 1000 bootstrap replicates. The procedure works identically as outlined in Sect. 3.4.2, except that for the Sonar dataset, the response variable follows a Bernoulli distribution, so that for model fitting (and refitting), we need to work with the binomial response

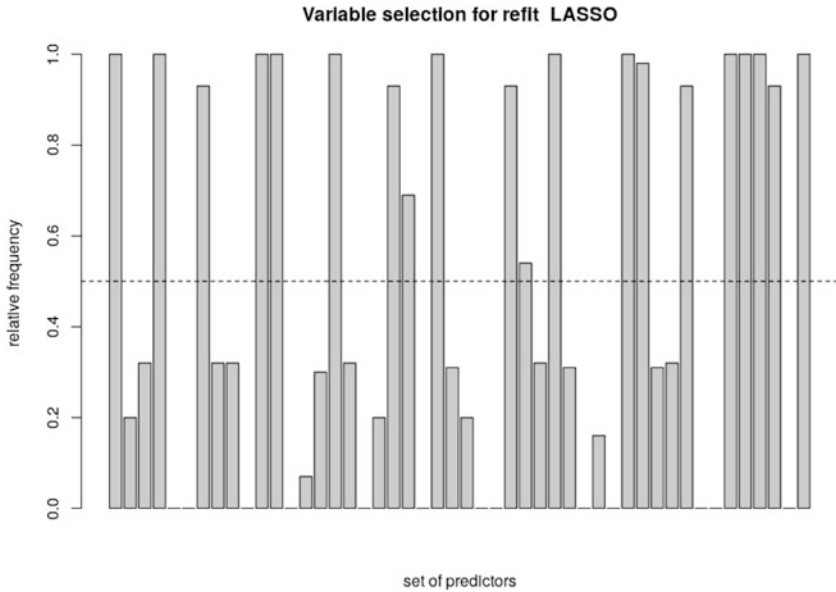


Fig. 3.13 Relative frequency of occurrence of variables, for refit-LASSO applied on the Sonar dataset

family instead of the normal distribution. The graph in Fig. 3.14 shows the bootstrap distribution of the estimated parameters.

3.5.2.3 Bayesian Approach

We obtained the posterior distribution of the parameters using the `MCMClogit` function from the `MCMCpack` [19] package in R. We took the Laplace priors for parameter estimation. We have taken 100,000 MCMC samples with a thinning interval length of 10 and a Metropolis tuning parameter set at 0.05, yielding 10,000 posterior samples for the assessment of the coefficient distribution. It can be seen that for the Bayesian approach the variability is almost same as that of bootstrap method (Fig. 3.15).

For a better comparison between each parameter estimation method, we have shown the standard errors for the coefficient estimates of each important variable in Fig. 3.16 indexed according to the refit-LASSO method.

Table 3.3 Summary of refit-LASSO for the Sonar dataset. The column “Estimate” gives the parameter estimates obtained after refitting the model using the selected variables. “Original” estimates refer to a (single) initial cross-validated LASSO execution as discussed in Sect. 3.5.1.1, and “Difference” refers to the difference between the refit-LASSO and the original estimates

Predictors	Estimate	Std. error	z value	$Pr(> z)$	Original	Difference
(Intercept)	-0.49	0.24	-2.01	0.04	-0.24	-0.24
V1	-0.72	0.33	-2.16	0.03	-0.12	-0.60
V4	-0.91	0.39	-2.32	0.02	-0.26	-0.65
V7	0.67	0.30	2.18	0.03	0.00	0.66
V11	-1.10	0.49	-2.24	0.02	-0.53	-0.57
V12	-0.34	0.41	-0.82	0.41	-0.25	-0.09
V16	1.05	0.34	3.14	0.00	0.29	0.76
V20	-0.88	0.57	-1.54	0.12	-0.03	-0.85
V21	0.25	0.57	0.44	0.66	-0.27	0.52
V23	-0.77	0.33	-2.35	0.02	-0.17	-0.60
V28	0.12	0.41	0.30	0.77	-0.10	0.22
V29	-0.63	0.48	-1.31	0.19	0.00	-0.63
V31	0.87	0.31	2.82	0.00	0.14	0.73
V36	1.04	0.58	1.80	0.07	0.58	0.46
V37	0.27	0.56	0.49	0.62	0.05	0.23
V40	0.34	0.33	1.04	0.30	0.01	0.34
V43	-0.03	0.46	-0.06	0.95	-0.07	0.04
V44	-0.80	0.58	-1.37	0.17	-0.14	-0.66
V45	-0.80	0.79	-1.01	0.31	-0.52	-0.28
V46	-0.06	0.64	-0.09	0.93	-0.02	-0.04
V48	-1.23	0.39	-3.16	0.00	-0.38	-0.85

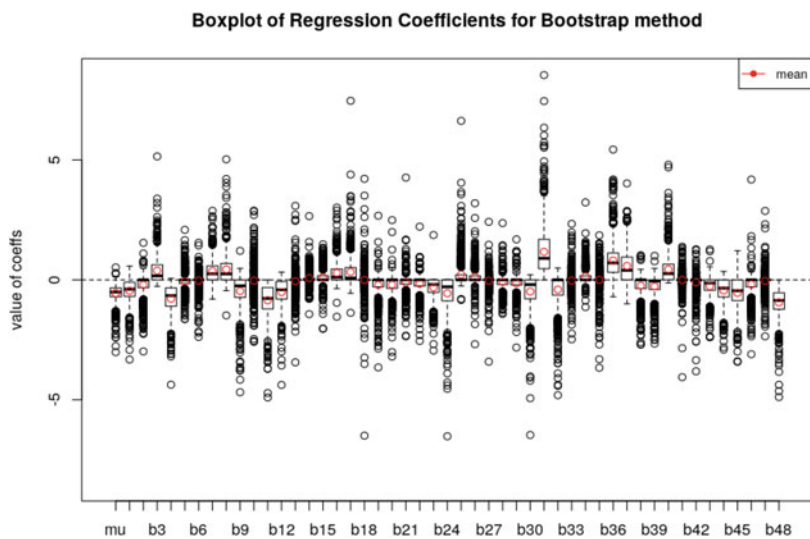


Fig. 3.14 Coefficient distribution of the bootstrap estimates for the Sonar dataset

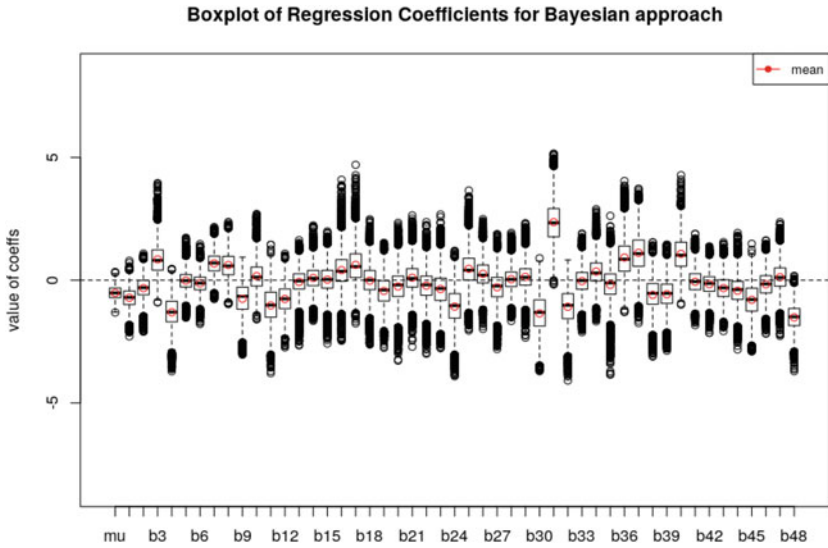


Fig. 3.15 Coefficient distribution of the Bayesian LASSO estimates for the Sonar dataset

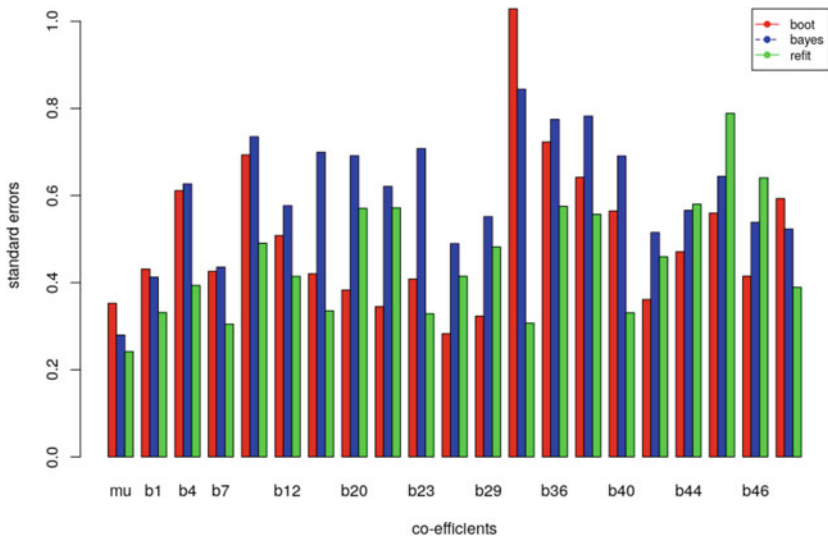


Fig. 3.16 Standard errors from different methods, for the logistic LASSO applied on the Sonar dataset

3.6 Conclusion

We have presented an overview over commonly used methods for uncertainty quantification in the context of l_1 -penalized linear or logistic regression, comprising refit, bootstrap, and Bayesian approaches.

We have illustrated these methods in the context of two datasets, both of which have some relevance for aerospace engineering: one dataset relating to the current Gaia space mission and another dataset involving the analysis of sonar signals.

For both modeling scenarios, we found good agreement of the parameter uncertainties obtained through the different methods. Standard errors of the bootstrap and refit methods agreed particularly closely, noting however the limitation of the latter to quantify uncertainty of inclusion as such. The Bayesian standard errors were of the same magnitude as their frequentist counterparts; however they tended to be larger and also did show some differences for specific parameters. For the Sonar dataset, the refit indicated sparser models than Bayes or bootstrap, which may appear unexpected at first glance but can be explained by the cut-off threshold of 50% which happened to be just above the relative frequencies of occurrence for many of the variables.

While the discussed uncertainty quantification methods are well-established and investigated for the linear model, this is less the case for the logistic model. This is not only reflected in the abundance of relevant literature, but also in the availability of statistical software. Since we had not been able to locate an implementation of the Bayesian logistic LASSO which could handle a model with 60 variables, we had to reduce this dataset from the start to 48 variables. We did so for all methods, to ensure comparability.

References

1. C.A.L. Bailer-Jones, The ILIUM forward modelling algorithm for multivariate parameter estimation and its application to derive stellar parameters from Gaia spectrophotometry. *Mon. Not. R. Astron. Soc.* **403**(1), 96–116 (2010)
2. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004)
3. L. Breiman, Better subset regression using the nonnegative garrote. *Technometrics* **37**(4), 373–384 (1995)
4. K. Das, M. Sobel, Dirichlet Lasso: a Bayesian approach to variable selection. *Stat. Modelling* **15**(3), 215–232 (2015)
5. N.R. Draper, H. Smith, *Fitting a Straight Line by Least Squares: Applied Regression Analysis* (Wiley, New York, 1998), pp. 15–46
6. B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression. *Ann. Stat.* **32**(2), 407–499 (2004)
7. J. Einbeck, L. Evers, C. Bailer-Jones, Representing complex data using localized principal components with application to astronomical data, in *Principal Manifolds for Data Visualization and Dimension Reduction*, ed. by A.N. Gorban, B. Kégl, D.C. Wunsch, A.Y. Zinovyev (Springer, Berlin, 2008), pp. 178–201

8. ESA science & technology: Gaia. <http://sci.esa.int/gaia>. Accessed 6 Feb 2018
9. L. Fahrmeir, G. Tutz, W. Hennevogl, *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer Series in Statistics (Springer, New York, 2001)
10. J. Friedman, T. Hastie, H. Hofling, R. Tibshirani, Pathwise coordinate optimization. *Ann. Appl. Stat.* **1**(2), 302–332 (2007)
11. J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33**(1), 1–22 (2010)
12. R.P. Gorman, T.J. Sejnowski, Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Netw.* **1**(1), 75–89 (1988)
13. R.B. Gramacy, monomvn: estimation for multivariate normal and Student-t data with monotone missingness (2017). R package version 1.9-7
14. I. Hacking, *The Emergence of Probability: A Philosophical Study of Early Ideas about Probability, Induction and Statistical Inference* (Cambridge University Press, Cambridge, 1975)
15. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics (Springer, New York, 2001)
16. T. Hastie, R. Tibshirani, M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability (CRC Press, Boca Raton, 2015)
17. F. Leisch, E. Dimitriadou, mlbench: machine learning benchmark problems (2010). R package version 2.1-1
18. A. Lykou, I. Ntzoufras, On Bayesian Lasso variable selection and the specification of the shrinkage parameter. *Stat. Comput.* **23**(3), 361–390 (2013)
19. A.D. Martin, K.M. Quinn, J.H. Park, MCMCpack: Markov chain Monte Carlo in R. *J. Stat. Softw.* **42**(9), 22 (2011)
20. Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st edn. (Springer Publishing Company, New York, 2014)
21. M.R. Osborne, B. Presnell, B.A. Turlach, On the Lasso and its dual. *J. Comput. Graph. Stat.* **9**(2), 319–337 (2000)
22. T. Park, G. Casella, The Bayesian Lasso. *J. Am. Stat. Assoc.* **103**(482), 681–686 (2008)
23. V. Ročková, E.I. George, The spike-and-slab Lasso. *J. Am. Stat. Assoc.* **113**(521), 431–444 (2018)
24. R. Tibshirani, Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Series B Stat. Methodol.* **58**(1), 267–288 (1996)
25. R. Tibshirani, Regression shrinkage and selection via the Lasso: a retrospective. *J. R. Stat. Soc. Series B Stat. Methodol.* **73**(3), 273–282 (2011)
26. M. Yuan, Y. Lin, On the non-negative garrotte estimator. *J. R. Stat. Soc. Series B Stat. Methodol.* **69**(2), 143–161 (2007)

Chapter 4

Reliability Theory



Daniel Krpelík, Frank P. A. Coolen, and Louis J. M. Aslett

Abstract Devices are of little use to us if they do not function properly, but whether they will function or not is subjected to uncertainty. Reliability theory studies the failure laws, i.e. constructs models and reasons with the chance that a device is functioning. Once we have obtained such models, we can take the reliability aspects into account during the design process.

This chapter introduces basics of mathematical reliability theory with emphasis on how can the reliability depend on design parameters.

Keywords Reliability · Survival analysis · System reliability · Design for reliability

4.1 Reliability and Risk

Causality is an essential principle which aids us to cope with the complexity of the world and, at least approximately, and to predict possible consequences of various actions. Nevertheless, since our understanding of the world comes from idealised models with limited scope of applicability, the derived predictions will usually deviate from the observed reality. As a result, we are generally not able to predict future events with absolute certainty. That is why we need to generalise our methods of reasoning to release ourselves from the shackles of binary logic and to include statements about possibilities and their various degrees of credibility, which allows us to rigorously address notions of “likeliness”, “chance” and “probability”.

D. Krpelík (✉)

Department of Mathematical Sciences, Durham University, Durham, UK

Department of Applied Mathematics, VŠB - Technical University of Ostrava, Ostrava, Czechia

e-mail: daniel.krpeлик@durham.ac.uk

F. P. A. Coolen · L. J. M. Aslett

Durham University, Durham, UK

© Springer Nature Switzerland AG 2021

M. Vasile (ed.), *Optimization Under Uncertainty with Applications to Aerospace Engineering*, https://doi.org/10.1007/978-3-030-60166-9_4

Reliability theory is a field at the interface of mathematics and engineering which primary interest is to evaluate whether a system (a device, policy, treatment, etc.) will behave as desired. Historically, since it cannot be predicted with certainty, a lot of interest was allocated into assessing the *validity* of the logical statement *the system will work*. The natural choice of the *validity* measure seems to be probability, since probability theory offers a consistent reasoning apparatus in which one can, deductively, from a set of basic assessments, derive probabilities of related statements similarly as in the familiar system of binary logic. But care must be taken to properly interpret the actual numerical values, probabilities, which this approach allocates to propositions. Probability theory is commonly used for assessing statements about both the frequency of events and the likeliness of occurrence of specific outcome in the next conducted trial. The first interpretation focuses on describing the sampling process, the anticipated relative ratio of occurrences of any particular trait of outcome and the aleatory uncertainty. If the predictive model is *well-calibrated*, the relative frequency tends to be *close* to the numerical value assigned to it by the model, where the closeness is understood in a limit of infinitely many trials, as in the law of large numbers. The second interpretation is also often termed *epistemic uncertainty*, since for any particular separate observation, an assigned numerical value of probability does not correspond to any observable quantity. The outcome of the next observation is a precise value, and we will perceive it as such, once the observation have been realised. Probability theory, here, serves as a tool to describe our state of knowledge, perception of *likeliness* of occurrence of an event, and allows us to reason about particular attributes of the future observation, including what actions we might take in order to improve the *chance* that the future observation will have desired properties, like “Is the system more likely to function if we use component A instead of component B?”. In the epistemic interpretation, the quality of a reasoning procedure manifests as an observable quantity through relative frequency of *correct* decisions in a series of repeated applications and the quality of the analytic methods rather than of the constructed models.

Both interpretations are relevant for applications of the reliability theory. Aleatory interpretation plays a role, e.g., for planning processes in which we assume that components will need to be replaced over time, and we need to schedule the maintenance and inspection policies (Sect. 4.5.2) or for the statistical quality control (Sect. 4.4.4). With good enough models, we can assess the long-time costs associated with operating our systems and also optimise the policies addressing their manufacture, maintenance and the logistical issues associated with the replacements. In such scenarios, failures are anticipated. Sometimes, we may discover that using a lower quality component may be beneficial from an economical perspective, leading to overall lower costs of the operation.

On the other hand, with some systems, usually the one-of-a-kind ones, we simply cannot afford them to fail. Some examples are the nuclear power plants, airplanes, residential buildings and many others. In these cases, we need to “ensure” that either the system works perfectly or we can detect an upcoming failure in time to mitigate its consequences. The issue is addressed by the so-called risk analysis which focuses on enlisting possible undesirable events and constructs measures aimed

at preventing them. Reliability analysis is used here for quantitative assessment of whether such a measure is adequate, but the underlying interpretation of probability is purely epistemic.

In this chapter, we will describe the basic question of reliability theory—predicting an occurrence of an event—with some further discussion on the engineering applications. Our choice of modelling tool is the probability theory, which has been standard in this field, although the magnitude of the uncertainties involved in some of the applications is better captured by imprecise probability models, which offer more degrees of freedom in modelling the available information (see Chap. 2). The issues will mostly be demonstrated on evaluating the probability that a system will complete its designated task once built and put into operation. The actual meaning behind the assigned numerical probabilities varies among applications and is, therefore, upon the particular analysts to translate it for their situation. For simplicity, we may, hereon, assume the frequency interpretation of probability measures; thus the probability of system functioning will mean that if we were to test “infinitely” many instances of the same system, reliability tells us what fraction of them will be functional. Nevertheless, if any quantity in a model is uncertain in an epistemic sense, the overall model inherits this interpretation and can further only be used to describe our degree of *belief* in the occurrence of an event.

In the second section, we will introduce the basic terminology and interpretation of the quantities used in reliability theory. In the third section, we will show how a problem can be decomposed into smaller parts when viewed as a system of components. In the fourth section, we will show some applications of statistics in reliability theory, hence how to answer some of the reliability-related questions on the basis of available observations. In the fifth section, we will show some methods for increasing reliability of systems by adding redundant components and how the system maintenance and other policies enable us to operate systems over long periods of time.

4.2 Mathematical Theory of Reliability

4.2.1 Structural Reliability

Our general aim is to construct a system which will function as desired. Once put into operation, the system will occupy a specific state $x \in \Omega_X$. Suppose that in the set Ω_X , we may further distinguish states which we label as being desirable, $\Omega_M \subset \Omega_X$, to represent what we actually *mean* by if a system functions. This might represent that the stresses on a bridge are smaller than its resistance so it will not collapse or that two planes pass at safe distance and will not crash, etc. But since the system is subjected to interaction with the real world, hence inherits its intrinsic uncertainties, our knowledge about the actual state will also be uncertain. Say we model it by a random variable X obtaining values in Ω_X . Now, instead

of precisely determining whether it is functioning (true or false), we must employ more sophisticated method to measure the *validity* of statements. Since we have decided to model uncertainties by the means of probability theory, we will measure the **reliability** of a system by the probability that the event $\{X \in \Omega_M\}$ occurs. The greater the probability becomes, the greater confidence we have that the system will actually work once deployed, or, in the frequency interpretation, the larger fraction of deployed systems will be functional.

In the context of system design, we are interested in selecting the “best” possible configuration for a system. Say that we may describe all the possible configurations by a design parameter $d \in \Omega_D$. Then, the actual state of the device may be viewed as a function of design parameters $\Omega_D \rightarrow \Omega_X$. If no uncertainties are present, we assess whether the system functions or not simply by assessing whether $\{x(d) \in \Omega_M\}$ is true or not and further restrict our admissible design space to a subspace for which the system would function, $\Omega_{D(M)} := \{d \in \Omega_D : x(d) \in \Omega_M\}$. But, due to the uncertainties, such crisp restriction of the design space is generally not possible. In such a case, the design parameters will specify random variables, representing the system state for different configurations, and for each of them, we may assess the probability that the system will function, the system reliability, $\text{Rel}(X(d)) := \text{Pr}(X(d) \in \Omega_M)$. The design problem, which generally aims to optimise also other performance measures over Ω_D , like the cost or performance, will have to take this into account by either restricting the design space to a subset with a priori selected reliability level, $\Omega_{D(M)} := \{d \in \Omega_D : \text{Rel}(X(d)) \geq \alpha\}$ for a selected level α , or by introducing another objective function to maximise the $\text{Rel}(X(d))$ and therefore necessarily lead to a multi-objective formulation. In order to assess the system reliability, one needs to be able to construct the probability model for the random variable $X(d)$ for any considered design parameters in Ω_D .

From the high-level perspective, and for simplicity, we will consider the state of a system as a binary random variable $X \in \{0, 1\}$, with 1 representing that the system functions, that $X \in \Omega_M$, and 0 otherwise. There is also a possibility to refine our model to include states of partial failure, or even several degrees of degradation, but we will omit that, for it would shift our concerns away from the basic reliability formulation toward general performance prediction.

4.2.2 Survival Analysis

A common property of real devices is their deterioration; their reliability will gradually decrease in time. But devices are usually required to function over the whole time periods, w.l.o.g. say the interval $[0, T]$. In order to take the time evolution into account, instead of a single random variable X , we need to investigate the whole stochastic process $X(t)$, representing the state of the system at time t , and reformulate the device mission event as $\{\forall t \in [0, T_M] : X(t) \in \Omega_M\}$. Note that this evolution would still be dependent also on the design $d \in \Omega_D$ as explained in the

previous subsection, making the system state a function of both t and d . We will omit the design parameter for the rest of this section.

Much interest in reliability theory lies in modelling this deterioration process [14, Sec. 6]. An intuitive way is to consider that a device depletes some intrinsic resources (or equivalently that it is accumulating a “wear out”) and define the failure state as such with these resources depleted. Let us consider a non-decreasing function $H(t)$, which models a cumulative depletion of some inner resource, and H^0 the amount of this resource available to the device. Then we consider the device functional at time t , if it has not yet depleted its inner resources, i.e. if $H(t) < H^0$. Equivalently, we can interpret this as the device not having reached a critical amount of “wear out”, e.g. accumulation of sediments or overall mass loss due to abrasion.

In order to provide an assessment of reliability, we need to model $H(t)$ and H^0 , which will generally both be uncertain. If these models are available (e.g. based on physics models), we can, again, employ probability theory to assess probability of the event of interest. These models may be available for specific problems (crack formation and abrasion, see [22] for more). If they are not available for the investigated system, reliability theory aims to provide ways of constructing them by the methods of statistical inference (some examples are shown in Sect. 4.4 and more can be found in any statistics textbook, e.g. [6]).

Let us consider a common scenario for a new device put into operation. If we assume that the device is functioning at time $t = 0$ (we try to assure this by post-production testing, but it is possible to generalise the methods for cases with so-called *hidden failures*), we can model the **time to failure** (TTF), the time when the device depletes its inner resources—thus it fails, as a non-negative random variable. As a random variable, the TTF can be described by its **cumulative distribution function** (CDF) $F(t)$ or, more commonly in the reliability theory context, its **survival function** $R(t)$.

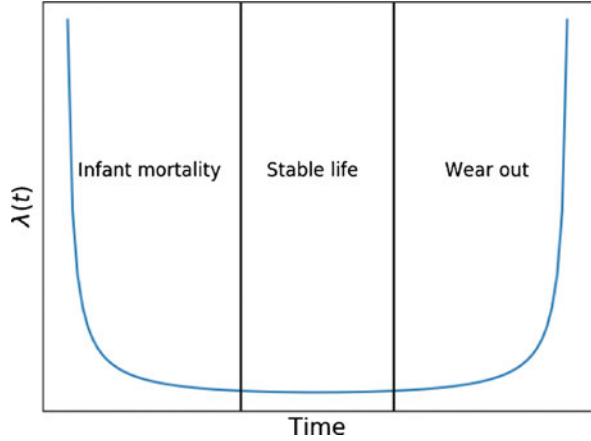
$$F_{\text{TTF}}(t) = Pr(\text{TTF} < t), \quad R_{\text{TTF}}(t) = 1 - F_{\text{TTF}}(t).$$

The advantage of modelling the TTF lies in the straightforward specification of the probability that a device will be operational over the mission time T_M ,

$$Pr(\text{device fulfils its mission}) = Pr(\text{TTF} > T_M) = R(T_M).$$

The most commonly used distribution to describe the TTF is the exponential one. This model assumes that the failures occur at random, regardless of how long the device has already been operational and what was its operational history. For the exponential distribution, $R(t) = \exp(-\lambda t)$, for a *failure rate* parameter λ . The exponential distribution is often used just for its mathematical convenience, although there are situations where its usage is justified, e.g. for modelling devices during the stable life phase (Fig. 4.1). Other distributions, which provide more flexible modelling options, are, e.g. Weibull, Cauchy, Log-normal or Gamma distributions. More about the basic mathematical models can be found in any introductory text in reliability theory, e.g. in [14, Ch. 3].

Fig. 4.1 The **bathtub curve** demonstrating the evolution of the failure rate of a standard device through its lifetime



An interesting contribution of the reliability theory, convenient to engineers, is the introduction of the failure rate function (also sometimes called the *hazard rate*) $\lambda(t)$. Failure rate describes an immediate failure probability:

$$0 \leq \lambda(t) := \lim_{h \rightarrow 0^+} \frac{Pr(TTF \in [t, t + h] | TTF > t)}{h \cdot Pr(TTF > t)} = \frac{f(t)}{R(t)}, \tag{4.1}$$

where the second equality is valid in cases of absolutely continuous CDFs, and $f(\cdot)$ denotes the probability density function (PDF) of the TTF.

Using Eq. (4.1), the failure rate is determined by the distribution of the TTF. The opposite is also true, as one can derive the TTF distribution from the failure rate as

$$R(t) = \exp\left(-\int_0^t \lambda(x) dx\right).$$

It can therefore be seen also as the rate of depleting the inner resources H^0 , as described in the section’s introduction. The failure rate also allows us to describe some qualitative properties of the failure laws. A general model of the evolution of the failure rate of a device over its lifetime is depicted in Fig. 4.1. During the first period, failures are mainly caused due to the faults of the manufacturing process (the infant mortality); in the second, the device experiences random failures due to the volatile nature of its environment (stable life); and in the last, the failures tend to be caused by wearing out of parts and components (wear-out phase).

These phases may be mixed during the device lifetime, and often just one is used to describe device failure law. Mathematically, a failure rate is a combination of the following phases:

- the failure rate is constant (stable life)—e.g. electrical components are judged to have constant failure rates.

- the failure rate is increasing (wear out)—e.g. mechanical components are subjected to abrasion, etc.
- the failure rate is decreasing (infant mortality)—e.g. software, as bugs are discovered and fixed during early stages.

4.3 System Reliability

One of the typical attributes of the contemporary world is its complexity. Every device and service available to us provides us with a building block, an opportunity to use it for constructing a new system with its own objectives. As a consequence, many of the devices and services that we rely upon are practically composed of many interconnected smaller *subsystems*. We can reflect this by our mathematical models and use the tools of probability theory to deduce the probability model for the system behaviour from the probability models for the behaviour of its components and their mutual interconnection. Exploiting the system structure also leads to significant savings of resources. If we were to assess the reliability of a complex system, like a space shuttle, by standard statistical methods, we would have to design an experiment in which we test (break) multiple copies of the same system. This would clearly lead to a vast waste of resources in the system design process. Identifying the system components will, instead, allow us to carry out cheaper experiments separately for those and even utilise our past experience with them. Nevertheless, the separate experiments would not allow us to learn about dependencies among the failure modes of components (common cause failures, cascading failures) which have to be addressed separately.

The way we carry out inference about complex systems may be decomposed into three major stages:

- Construct a model of the system consisting of components (subsystems) reflecting dependencies between states of the components and the state of the system as a whole.
- Gather data and carry out inference about components, based on, for example, statistical methods or expert elicitation procedures.
- Integrate the acquired models of the components behaviour with the model of their influence on the system behaviour in order to obtain a model for TTF of the whole system.

The first stage is a domain of engineers who have to specify the system topology and carry out risk analyses to identify potential modes of failure and describe how the system operates. Component models can be obtained by statistical methods and are also often included in the component specification in the case of sub-contracting, although, often, only partial specifications are available, in this case like first and second moments of the component failure laws. In this section, we will further focus on the first and the third part of the inference process, on how to integrate the acquired information.

4.3.1 Structure Function

Description of the dependency among the state of the system and the states of its components can be provided by a deterministic function. For each possible combination of components states, functioning or failed, we determine whether the system is functioning or not. The uncertainty of the system state will then arise solely due to the uncertainties about the states of its components. Let us denote the (deterministic) state of the system as $x_S \in \{0, 1\}$ and a vector of states of its N components as $\vec{x} \in \{0, 1\}^N$. We will restrict ourselves to systems with binary components, since it covers many practical scenarios. This restriction can be dropped if necessary to describe any relationship among the system and its components but would lead to more complicated mathematical models. We define the (deterministic) **structure function** as a function φ which maps states of the components onto the state of the system; thus $x_S = \varphi(\vec{x})$. The structure function is therefore, in our restricted case, a Boolean formula on N variables (an example is given in Table 4.1).

If an uncertainty about the component states is present, first, we model the states of the components by a random vector \vec{X} . Note the capital letter representing random variables as usual in the probability theory literature. The state of the system will inherit the uncertainty from the states of its components and, in the model, becomes a binary random variable X_S . We can now assess the system reliability by taking the expectation of $\varphi(X)$,

$$\text{Rel} = Pr(X_S = 1) = \mathbb{E}\{\varphi(\vec{X})\} = \sum_{\vec{x} \in \{1,0\}^N} \varphi(\vec{x}) Pr(\vec{X} = \vec{x}). \quad (4.2)$$

Table 4.1 An example of the structure function for a $N = 4$ component system

\vec{x}				$\varphi(\vec{x})$
0	0	0	0	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	1
0	1	1	0	0
1	1	1	0	1
0	0	0	1	0
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

The reliability of a system can also be expressed by the reliability function $h : [0, 1]^N \rightarrow [0, 1]$, which directly models the relation between a vector representing probabilities that each individual component functions and probability that the system functions. For example, for a serial system (all components have to function to consider the system to be functioning) with $N = 3$ components with $p_i := Pr(X_i = 1)$ being the reliability of component i , it holds that $Pr(X_S = 1) = h(p_1, p_2, p_3) = p_1 \cdot p_2 \cdot p_3$.

The structure function, as defined here, is dependent only on the current states of the components and, thus, allows us to separate static structure dependencies from temporal evolution of component states as described in Sect. 4.2.2. The same applies for the reliability function, which only depends on the probability that components function at a given time instance. Generalisations are possible, but the actual mathematical model is dependent on the investigated scenario. Nevertheless, even in our restricted case, the evaluation of a system reliability has exponential complexity. It would require us to sum over all the elements of the state space ($\sim 2^N$). The reliability function is also exponentially complex to construct but may be later used multiple times, e.g. for reconstruction of temporal evolution of system state (the survival function) or in the problems of statistical inference, and make these tasks tractable.

The structure function can be generally described by a table, prescribing the state of the system to every possible configuration, but such a table would be impractical to construct, work with and inspect for any system of realistic size, because the number of rows grows exponentially with the number of components. There exist several alternative ways to specify the structure function. These enable us to present the structure function graphically which also allows us to analyse it qualitatively by the tools and notions of the graph theory.

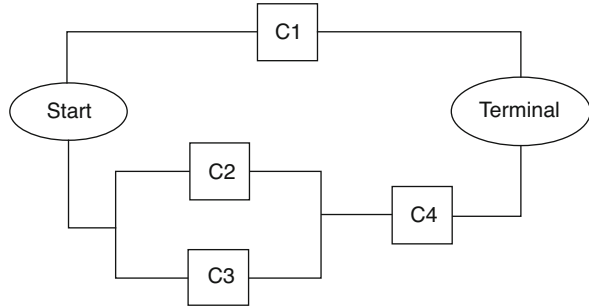
4.3.2 Graphical Models

4.3.2.1 Reliability Block Diagrams

Reliability block diagrams (RBDs) capture how the system components are connected [17, Ch. 5]. They constitute a natural way for modelling systems whose function is related to various kinds of transportation and communication (railroads, computer networks, etc.) but can be generally used to depict any structure function.

For traffic networks, communication networks or also the power networks, an RBD describes the network topology and allows us to easily construct structure functions for classes of problems addressing the so-called k -terminal network reliability. For these problems, we define that a system with N components functions if the “ k ” pre-specified components are connected through nodes corresponding to functioning components. But RBDs do not need to refer to anything physical and can be used just as a graphical description of the system structure function.

Fig. 4.2 An example of RBD equivalent to the structure function in Table 4.1



An example of an RBD, equivalent to the structure function in Table 4.1, is shown in Fig. 4.2. It is a 2-terminal network, where the nodes that need to be connected in order to consider the system functional are denoted “Start” and “Terminal” and do not correspond to any physical component of the system. Components of the system are represented by nodes “C1-4”, where the number indicates the column in Table 4.1 corresponding to the respective component.

4.3.2.2 Fault Trees

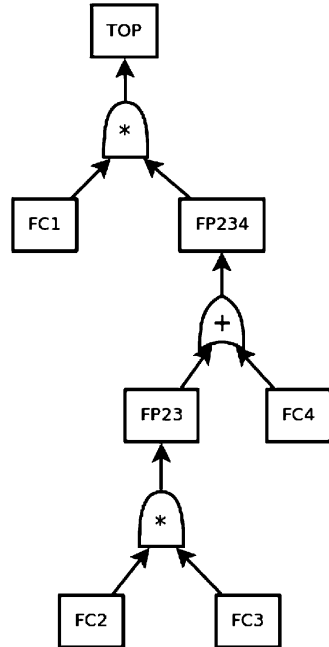
Another way to obtain a graphical description of a system is by the means of a **fault tree analysis** (FTA) [11], [14, Sec. 12]. Here, the aim is to, recursively, describe which causes lead to an event being deconstructed. We start by defining a *top-level event*, the event of system failure, and investigate which causes trigger it. The causes do not have to be directly elicited in terms of states of singular components. The algorithm recurs to find the causes of these causes and so on as far as we wish up to so-called *terminal events*, the finest refinements of the state space. In order to assess the reliability of the system, it is necessary just to describe the probabilities of the occurrence of the terminal events. The state of the whole system is then assessed through a structure function $\varphi(\vec{e})$; arguments of which are vectors denoting the occurrence or states of the terminal events s.t. $e_i \in \{0, 1\}$, which events are selected to be the terminal ones is arbitrary, up to an analyst, and they, again, do not need to be states of the system components.

A fault tree represents a hierarchical Boolean formula. The actual fault tree is composed of events and gates. The events are events in the sense of probability theory, subsets of the sample space and logical statements (binary). The gates are Boolean functions (e.g. AND, OR, K-of-M and NOT) used to describe how the combination of events induces a macro-event higher in the tree hierarchy.

Once again, a general structure function of a system may be transformed into a fault tree, which would provide its graphical depiction. An example of a fault tree corresponding to the structure function in Table 4.1 is shown in Fig. 4.3.

The fault tree methodology provides a way for conducting risk analysis of general systems where we cannot construct the structure function nor sometimes even elicit

Fig. 4.3 A fault tree of example system. “TOP” represents failure of the system, “FCx” are failures of respective components (terminal events) and “FPx” are macro-events



all the components and events influencing the state of the system. The reason is that we advance from the top event to arbitrary depth. As an example take an automobile. The top event would be that the automobile does not drive you to your final destination. What could cause that? Maybe there is no gasoline in the tank, or the ignition malfunctions, or the engine is jammed. Well, what could cause the engine to be jammed? Maybe it is due to a mechanical displacement, or the oil was not replaced, or... Well, what could cause the mechanical displacement? ... And so on, up to the desired level of detail.

The tree, which models the relations between the events and the causes of these events, may be constructed from expert knowledge, or from fault logs obtained from deployed systems. The FTA can also easily consider external factors leading to failure. With the model of dependencies available, we only need to assess the probabilities of the considered terminal events, not necessarily create stochastic models for states of all the components (and the environment).

4.3.2.3 Bayesian Networks

The structure function assumes that the dependencies between the components and the system states are precisely known, but this may not always be the case. It may be that we did not reach necessary depth when constructing a fault tree to ensure unique relationship between an event and its triggers. As an example take a railway trip. One of the trains might be delayed and you miss your connection, but you still

might be lucky enough to encounter a helpful railroad clerk who will direct you to an alternative connection or not.

To model such situations, the structure function might be naturally generalised to include uncertainties about the dependencies among the states of the events in the fault tree simply by stating the probability of the event obtaining based on the states of the events lower in the tree hierarchy. A (graphical) tool used to depict these models is known as the **Bayesian network** (BN) [5], [13, Ch. 8]. The construction may be done by the FTA, but now we do not formulate the dependencies between a macro-event and its causes by Boolean functions but as conditional probabilities. This is, of course, a much more challenging task, but it also provides an advantage. It allows us to work with a less detailed models, since we need not to advance the tree construction up to the level in which all the relations would be deterministic, and these conditional probabilities can be inferred by statistical methods (also by robust statistical methods [3, Ch. 9]). Thus the main role of the FTA would be to elicit the relevant events and the assumptions on the conditional independence. Once the BN is constructed and the stochastic models are provided, the system reliability may be assessed, deductively, according to the theory of probability. For \vec{X} denoting the random vector of system components' states:

$$Pr(X_S = 1 | \vec{X} = \vec{x}) = \sum_{y_1, y_2} Pr(X_S = 1 | X_A = y_1, X_B = y_2) \cdot Pr(X_A = y_1, X_B = y_2 | \vec{X} = \vec{x}),$$

where X_A, X_B are the only macro-events such that the state of the system is conditionally independent from component states given X_A, X_B according to our structural assumptions.

The state of the system can be assessed recursively by marginalising over \vec{X} .

$$Pr(X_S = 1) = \sum_{\vec{x} \in \{0,1\}^N} Pr(X_S = 1 | \vec{X} = \vec{x}) Pr(\vec{X} = \vec{x}).$$

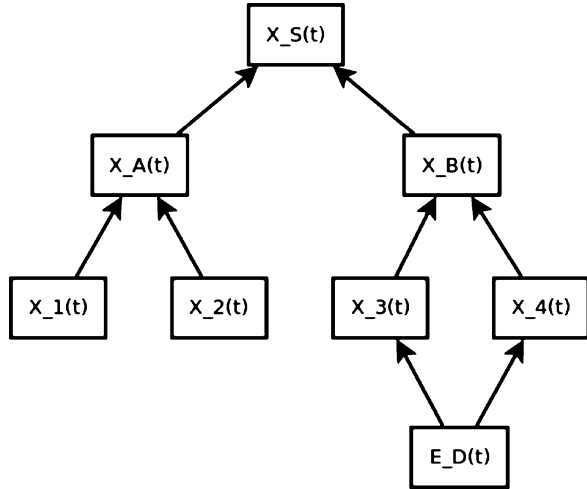
Bayesian networks can also be used to model dependencies among the component failures, e.g. common cause failures, where some external disturbance might affect multiple components at the same time. In such a case, component reliabilities may be specified as conditional on the occurrence of this disturbing event, e.g.

$$Pr(X_i(t) = 1 | E_D(t) = 0) = R_i(t)$$

$$Pr(X_i(t) = 1 | E_D(t) = 1) = 0,$$

for some disturbing event $E_D(t)$. In this scenario, the disturbing event would surely render the component failed. In order to assess the overall system reliability, the probability of occurrence of this disturbing event also has to be specified.

Fig. 4.4 Example of a Bayesian network with two macro-events X_A , X_B and an event E_D disturbing components 3, 4



An example of a graphical BN is shown in Fig. 4.4. For each of its nodes, a probability table conditional on its predecessors (unconditional for the terminal events) has to be specified.

4.3.3 Phased Missions

Some real systems do not operate under the same conditions and with the same functional requirements during their whole lifetime, and we might be able to identify different phases of their missions. The physical system may remain the same over these phases, but the functionalities we require it to provide, or the loads exerted upon the components may differ among these phases. Such scenarios are known in the literature as **phased mission systems** (PMS) [10]. An example might be an aircraft journey, where the aircraft must take-off, cruise along the flight path and, finally, land again.

The modelling is performed in two basic steps. First, we need to identify different phases, and for each of those we construct a model describing what constitutes a successful operation in this phase. These models may be specified by fault tree or RBD models. Then we need to link the models of all the phases together. If the phases are specified by fault trees, this linking will result into a single extended fault tree characterising the whole mission, similarly with the RBDs. In both cases, the following treatment is similar to that introduced earlier but with some specifics which need to be taken into account (Fig. 4.5).

A mission is considered successful if the system did not fail in any of its phases. From (monotone) structure function point of view, this means that for each time, which denotes the end of a mission phase, a milestone, the system must be functional

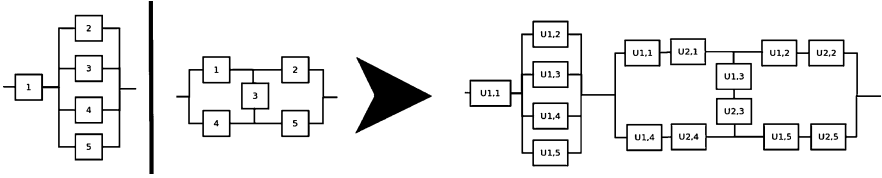


Fig. 4.5 Transformation of RBDs of mission phases into a RBD of a phased mission according to the Esary’s identity [10]. The events $U_{i,j}$ represent the conditional events that a component j does not fail at phase i given that it is functioning at its beginning. The “Start” and “Terminal” nodes are omitted

at that time. The monotonicity assures that the system was functional also during the whole phase. For example, for a mission with K phases and milestones t_1, \dots, t_K , the joint mission structure function is given by

$$\varphi_{mission}(\vec{X}(t_1), \dots, \vec{X}(t_K)) := \prod_{i=1}^K \varphi_i(\vec{X}(t_i)),$$

where $\varphi_{mission}$ represents the structure function of the whole mission (defined as $\varphi_{mission} : \{0, 1\}^{N \cdot K} \rightarrow \{0, 1\}$ for an N component system) and φ_i are structure functions in the respective phases ($\varphi_i : \{0, 1\}^N \rightarrow \{0, 1\}$).

Phased mission models can also be used for an on-line decision making during the mission execution. Once a model of the mission is constructed, we may not only assess the probability of successful completion of a mission but, in case we have modelled them, also the probabilities of completion of mission deviations. This may be useful in case some disturbances occur, which would endanger the mission’s completion. In such cases, we may quickly assess risks of possible alternatives and alter the mission, respectively [1, 2].

4.3.4 Signatures

An important tool for reliability assessment is the structure function, may it be specified by a RBD, FTA, BN or PMS. One problem with structure functions is their high dimensionality in practical scenarios (exponential in number of components) which turns any following reliability analysis into a computationally expensive process. *Signatures* allow us to overcome this problem by providing alternative descriptions of a system in a lower dimensional space, its summary, which is also often able to separate the mathematical term coming from system structure from the one corresponding to components’ TTFs.

Given a probability space, we can express the probability of any event via the law of total probability. Let us have an event S , that the system is working, and an

arbitrary decomposition of the space of component states $\{0, 1\}^N$ into disjoint sets D_1, \dots, D_k . The probability of event S can then be expressed as

$$P(S) = \sum_i P(S|D_i)P(D_i). \quad (4.3)$$

The signatures we describe in this section both rely on this formula but differ in the choice of the underlying decomposition.

The original **system signatures** were proposed by Samaniego [20] and celebrated successful applications in the system reliability analysis and system structure optimisation. On the other hand, they could only be applied to systems with components with independent identically distributed (i.i.d.) lifetimes, which is overly restrictive for many practical scenarios, since most of the systems are composed of heterogeneous components. This limitation was overcome by the introduction of **survival signature** [7] by Coolen and Coolen-Maturi which allows us to model systems with multiple types of components.

4.3.4.1 System Signature

System signature is introduced for systems composed of components with i.i.d. TTF. The i.i.d. requirement either restricts us to analyse systems consisting of multiple instances of the same component or to systems for which we assume that the other components are totally reliable (cannot fail). Nevertheless, many practical systems can still be analysed using this methodology, like traffic networks, telecommunication networks, computer components . . .

The system signature is defined as a discrete probability vector q_1, \dots, q_N , where q_i denotes the probability that the i -th component failure will result in the failure of the system. The expression system reliability can be simplified into

$$P(\text{TTF}_{\text{sys}} > t) = \sum_{i=1}^N q_i P(\text{TTF}_{(i:N)} > t),$$

where $\text{TTF}_{(i:N)}$ denotes the i th order statistic (a random variable describing the probability distribution of i th failure time in the sample of size N). In the i.i.d. case,

$$P(\text{TTF}_{(i:N)} > t) = \sum_{r=N-i+1}^N \binom{N}{r} [1 - F(t)]^r [F(t)]^{N-r},$$

where F is the common CDF for the component lifetimes.

Samaniego has shown that the system signature may serve as a way of comparing systems. He provides theorems about how different stochastic orderings of system

signatures implies stochastic orderings of system TTF [20]. This enables us to define system optimisation problems as problems of finding systems with optimal signatures, although not each signature correspond to a physical system.

4.3.4.2 Survival Signature

An extension to system signatures may be made for systems consisting of multiple types of components. In this scenario, we assume that the TTFs of components of the same type are exchangeable (i.i.d. implies exchangeability). This allows us to model more scenarios than the system signature (e.g. network system with both switch-boards and transmission ducts). The scenario in which the TTF of each of the components is different is also included as an extreme case.

Let us assume that we have a system with K distinct component types and denote G_j the set of components of type j and M_j the number of components of type j in the system. We can introduce a natural decomposition of $\{0, 1\}^N$ into $D_{\vec{l}}$, where $\vec{l} \in \otimes_{i=1}^K \{0, 1, \dots, M_i\}$ is a multi-index, and $D_{\vec{l}} := \{\vec{x} \in \Omega_X : \forall j : \sum_{i \in G_j} x_i = l_j\}$. This corresponds to a decomposition into disjoint sets $D_{\vec{l}}$ where for each component type j exactly l_j components are functioning. The probability $P(S|D_i)$ may be viewed, due to the exchangeability assumption, as the probability of success in a Bernoulli trial (the number of favourable events divided by the number of all the possible events) and may be derived from the structure function as

$$\Phi(\vec{l}) := P(S|\vec{X} \in D_{\vec{l}}) = \frac{|\{\vec{x} \in D_{\vec{l}} : \varphi(\vec{x}) = 1\}|}{|D_{\vec{l}}|} = \left[\prod_{i=0}^K \binom{M_i}{l_i}^{-1} \right] \sum_{\vec{x} \in D_{\vec{l}}} \varphi(\vec{x}).$$

The mixing probability, $P(D_i)$ from Eq. (4.3), is then

$$P(\vec{X} \in D_{\vec{l}}) = \prod_{i=0}^K \binom{M_i}{l_i} [P(X_i = 1)]^{l_i} [P(X_i = 0)]^{M_i - l_i}.$$

The survival function of the system is therefore separated into a time dependent (component reliability) and a time independent (system structure survival signature) factors, and

$$P(\text{TTF}_{\text{sys}} > t) = \sum_{\vec{l}=0}^{(M_1, \dots, M_K)} P(S|\vec{X} \in D_{\vec{l}}) P_t(\vec{X} \in D_{\vec{l}}).$$

If the TTF distribution of the components is independent on all the other components, the relation simplifies into

$$\begin{aligned}
P(\text{TTF}_{\text{sys}} > t) &= \sum_{\vec{l}=\vec{0}}^{(M_1, \dots, M_K)} \Phi(\vec{l}) \prod_{k=1}^K \left[\binom{M_k}{l_k} [P_t(X_i(t) = 1)]^{l_k} [P_t(X_i(t) = 0)]^{M_k - l_k} \right], \\
&= \sum_{\vec{l}=\vec{0}}^{(M_1, \dots, M_K)} \Phi(\vec{l}) \prod_{k=1}^K \left[\binom{M_k}{l_k} [1 - F_k(t)]^{l_k} [F_k(x)]^{M_k - l_k} \right].
\end{aligned}$$

Both the system and survival signatures allow us to greatly reduce the amount of storage necessary to describe a system and, subsequently, to analyse it. It is still exponentially expensive to calculate the signatures from a structure function, but, as with the reliability function, it is enough to carry out this computation only once and, possibly, to communicate the system specification itself solely via the signatures. Both of the signatures act as system summaries—there is no one-to-one correspondence between the respective classes of systems and their signatures—which has a further possible benefit for manufacturers in masking the actual system topology while still allowing to analyse and compare systems' performance by sub-contractors and researchers.

4.4 Statistical Inference in Reliability

As with other stochastic models, we can use the tools of mathematical statistics in order to *infer* the failure distributions from (mostly, but not only) empirical evidence. From observations of the device behaviour in the past, we can estimate possible behaviours in the future. The random variable of interest is the TTF, and the data from which we infer are usually the observed TTF of tested components. Since we demand the devices to operate for large periods of time (years and longer), it takes significant amount of time to collect the data from real experiments, because we need to wait for the observed devices to fail. For this purpose, special methodologies have been developed in reliability theory to help us overcome this difficulty.

4.4.1 Censored Datasets

In many cases, the experiment collecting observations of the failure times needs to be terminated before all the devices have yet failed. We could discard those units for which the failure had not occurred during the test in order to proceed with the analysis, but in such a case, we would lose a lot of acquired information, and it would also lead to incorrect conclusions because of the omission of the evidence for longer lifetimes. For these scenarios, the notion of **censored data** was introduced in order to build a theory of how to utilise all the available information and avoid

possible systematic bias caused by omitting part of the information [14, Sec. 8.4], [21, Sec. 5.4]. Such data may originate either due to the experimental design, i.e. we terminate the experiment when a pre-specified time T_{lim} has passed or when a certain pre-specified number of failures k_{lim} has been observed or because of random influences, e.g. failures due to a different cause than the one analysed or losing the track of the statistical unit (common to clinical studies).

In reliability theory, a common type of censoring is so-called **right censoring**, where we have terminated the experiment before all the devices have failed. Here, we can combine the censoring times with knowledge that the (not manifested) TTF is greater than the censoring time. In order to construct a (precise) likelihood which could be used by standard statistical procedures, an additional assumption needs to be made. The one usually used is that of a **random censoring** mechanism which states that the censoring time is stochastically independent of the failure time. One may imagine how such an assumption might be violated, for example, in medical survival studies, where the approaching failure may make the patient to reconsider his participation in the study. We would need to propose different censoring models based on the nature of the observations.

Let us assume that we have a set of independent statistical units and have observed a collection of failure times $\{t_1, \dots, t_n\}$ and also a collection of right censoring times $\{c_1, \dots, c_e\}$. The censoring times denote the supremum time for which we know the unit has not yet failed, but the exact time of failure is not known. Assuming the random censoring, the likelihood function, will take the form

$$\mathcal{L}(\theta; \vec{x}, \vec{c}) = \left(\prod_{i=1}^n f_{\theta}(t_i) \right) \left(\prod_{i=1}^e R_{\theta}(c_i) \right),$$

where f_{θ} and R_{θ} are the PDF and the survival function, respectively, indexed by the distribution family parameter θ . The inference about the distribution index θ may then be carried out by both frequentist and Bayesian methods.

4.4.2 Accelerated Life Testing

Another way to decrease the necessary experimental time is **accelerated life testing** (ALT) methodology [14, Sec. 8.5], [15], [21, Sec. 5.8]. The core of the method lies in the idea of exposing devices to harsher conditions in which they will deteriorate faster. In order to infer the distributions of the TTF in the working conditions, we must also choose a model for the deterioration speed-up. This model will provide us with the means of transforming the failure times at higher stress levels to the working conditions. This transformation model itself may be uncertain, dependent on some parameters. If that is the case, these parameters also have to be estimated during the inference process.

In the inferential scenario, we thus observe several failures from multiple levels of stress. Usually, the least amount of observations comes from the level with working stress conditions, because the devices tested on this level are assumed to deteriorate by the slowest rate. All the observations are then joined together in by single likelihood function for the model parameters; the model parameters are inferred, and the desired lifetime distribution for the working stress level is derived from them.

Accelerated life testing is thus among the methods which enabled practical testing of highly reliable components. It is dependent on modelling the underlying physical process which leads to the acceleration of the deterioration process. Furthermore, the model of acceleration is also inferred and may be utilised in design optimisation where we could investigate how the design parameters influence the working conditions of the device and, therefore, its reliability.

We will show how the ALT can be formulated and solved for two commonly used transformation models based on independent observations.

The **Arrhenius law** originated for describing how the transition rates changes for chemical reactions based on the environmental conditions. It may be used as a transformation model if we include an assumption that the lifetime distributions on each level are exponential, with constant failure rate, because it prescribes the relation directly between the failure rates on different levels. Let us parametrise the exponential PDF with its failure rate so that

$$f(t|\lambda) = \lambda \exp(-\lambda t).$$

Then we can link the mean times to failure at different levels by the Arrhenius law.

$$\mu(V) = C \exp\left(\frac{B}{V}\right),$$

where C, B are model parameters which need to be estimated; V is a physical observation describing the stress upon the component (e.g. electric potential, temperature, ...); and $\mu(V)$ is the mean TTF at level V . The failure rate on each level can thus be obtained by taking the reciprocal value $\lambda_V = \frac{1}{\mu(V)}$. Since we can now precisely specify the distribution at each level conditional on the knowledge of model parameters B, C , we can also construct the likelihood function for the model parameters B, C conditional on the observed values. That will take the form (with the independency assumption)

$$\mathcal{L}(B, C; \vec{t}, \vec{v}) := \prod_V \prod_i f(t_{V,i} | \lambda_V),$$

where $t_{V,i}$ denotes i -th observation on level V , $f(\cdot | \lambda_V)$ is the lifetime pdf on level V , \vec{t} are the observed failure times and \vec{v} the respective levels on which the observation had been made (Fig. 4.6).

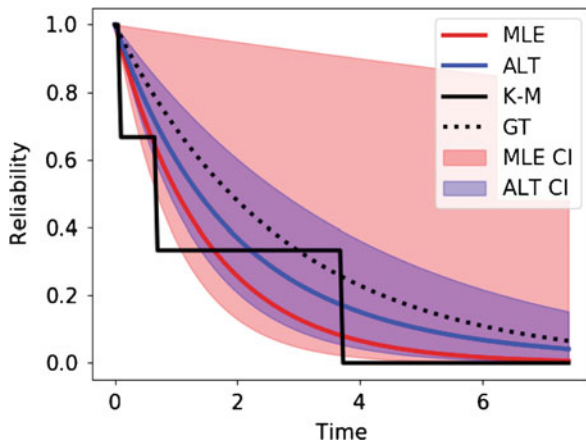


Fig. 4.6 An example of an ALT inference with dependency modelled with the Arrhenius law on 3 level with $(C, B) = (1, 1)$ and amount of observations at respective levels being 3,10 and 25. Curve “GT” represents the sampling distribution for the base level, and “K-M” is a Kaplan–Meier estimate based on samples from the base level. Results of the inferences are shown by curves “MLE”, for the maximum likelihood estimate based on the samples from the base level, and “ALT”, based on the Bayesian inference described in the current subsection. A confidence and credible intervals respectively are depicted as the shaded areas

Another model used to transform the observations from one level (external conditions setting) to another is the **power-Weibull** model. Here we assume that the distribution of the lifetime at all levels can be modelled by the Weibull distribution. An important factor is that the shape parameter needs to be the same in each of the stress levels, because the contrary would signify an introduction of new types of failure modes; thus failures which would not naturally occur in the working conditions and thus bias our inference. The PDF of the Weibull distribution, parametrised by shape β and scale α , is

$$f(t|\alpha, \beta) = \frac{\beta}{\alpha} \left(\frac{t}{\alpha}\right)^{\beta-1} \exp\left(-\left(\frac{t}{\alpha}\right)^\beta\right)$$

The transformation from level i to level j might be specified for the scale parameter α as

$$\alpha_j = \alpha_i \left(\frac{V_i}{V_j}\right)^p,$$

where p is another model parameter which will need to be estimated aside from the common shape parameters β and α_{V_0} , the scale parameter on level V_0 . From these three, we can uniquely determine the lifetime distribution for any stress level thus also construct the likelihood function necessary for the inference of (p, β, α_{V_0}) .

Once we obtain the posterior distribution for the model parameters $((B, C)$ or (p, β, α_{V0})) (in the Bayesian framework), we can propagate it directly to obtain the distribution for the TTF at the working level,

$$\begin{aligned} Pr(\text{TTF} < t) &= \int_0^t \left[\int_{\Omega_\theta} f(x|\theta) d\pi(\theta|\vec{t}, \vec{v}) \right] dx, \\ &= \int_{\Omega_\theta} F(t|\theta) d\pi(\theta|\vec{t}, \vec{v}). \end{aligned}$$

where θ represents the parameters of the chosen model, and $F(\cdot|\theta)$ is the lifetime CDF on the working level.

4.4.3 Proportional Hazards Model

Sometimes, it may be our desire to determine the influence of other available characteristics on the lifetime distribution. The means of inferring such dependency require us, similarly as in the case of the ALT, to choose and include a model of this dependency into the likelihood function. Assume that we observe a series of failure times $\vec{t} = (t_1, \dots)$ and also, for each of the statistical units, some value(s) \vec{x}_i representing their additional attributes. Now, we are looking for a mapping which would prescribe the lifetime distribution for any new unit with attributes \vec{x}' .

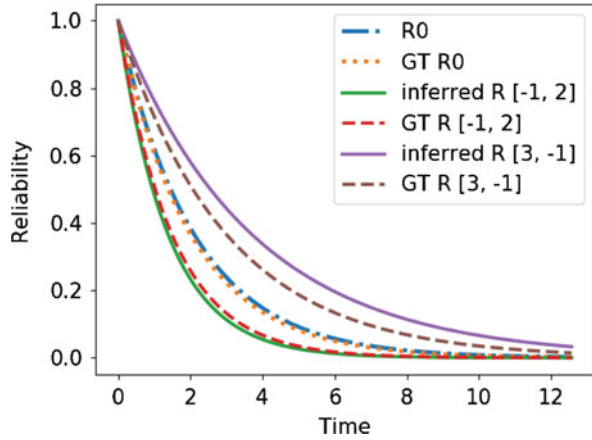
In the statistics literature, a lot of work has been focused on (generalised) linear models. Such a model was also introduced for solving the above mentioned problem by Cox, called after him the **Cox's proportional hazards model** [9]. As mentioned in Sect. 4.2.2, a (well-behaved) lifetime distribution may be uniquely specified by its failure rate function $\lambda(t)$. Hence Cox has proposed a linear model for the logarithm of the failure rate function in which, for a vector of d additional attributes $\vec{x} = (x_1, \dots, x_d)$, the model of the failure rate function takes the form

$$\lambda(t|\vec{x}) = \lambda_0(t) \exp \left[\sum_{j=1}^d \beta_j x_j^i \right],$$

where $\lambda_0(t)$ is a base-line failure rate function which may be inferred later and is common for the whole population, and $\beta = (\beta_1 \dots \beta_d)$ are model parameters which are to be estimated and which capture the influence of the covariates \vec{x} (Fig. 4.7). Cox's model allows us to determine influential factors and the nature of the influence also without the need to infer the base failure rate $\lambda_0(t)$ at all, which is why it is often used in bio-statistics to test hypotheses about sensitivity to a factor variations.

For the full inference, first, we decompose the sampling distribution into a factor modelling the chance of observing the failures at specified times and a factor modelling the conditional probability of observing them in specified order given

Fig. 4.7 An example of survival functions for units with varying covariates inferred via Cox’s proportional hazard model. GT stands for “Ground Truth”, the models from which the observations were generated, “ R_0 ” and “inferred $R[x, y]$ ” denote inferred survival functions, and the vector $[x, y]$ stands for values of the covariates



when the failures had occurred. The latter conditional distribution will turn out to be dependent only on the vector $\vec{\beta}$ and can therefore be solved separately. Once we infer the coefficients β , we may proceed with the inference for λ_0 .

For the first part, we condition upon the observed failure times \vec{t} and construct a *conditional* likelihood for the model parameters β . While conditioning upon the observed failure times, we are only interested in that if a failure happens, what is the probability that it is unit with covariates \vec{x} . Hence we can specify a partial likelihood for each of the failed units i as

$$\mathcal{L}_i(\vec{\beta}) = \frac{\lambda(t_i|\vec{x}^i)}{\sum_{j:t_j \geq t_i} \lambda(t_i|\vec{x}^j)} = \frac{\exp(\vec{x}^i \vec{\beta})}{\sum_{j:t_j \geq t_i} \exp(\vec{x}^j \vec{\beta})},$$

where the normalisation is carried over all the units *at risk* at time of the failure of the i -th unit. This treatment allowed Cox to also apply his method for censored observations. The censored observations come into play via the normalisation constant by decreasing the amount of units at risk at observed failure times. The equations for conditional likelihood remain the same in such case.

Once the partial conditional likelihood is specified for each of the observed failure times, the full conditional likelihood may be obtained (under the assumption of independence) by taking their product

$$\mathcal{L}(\vec{\beta}) = \prod_i \mathcal{L}_i(\vec{\beta}).$$

Once the β coefficients are inferred, we may use the result to specify the likelihood for $\lambda_0|\vec{\beta}$. Note that $(\lambda_0, \vec{\beta})$ define the failure rate function, thus also the survival function and the likelihood. Cox [9] used some simplifications for the inference of λ_0 . The β were estimated by a maximum likelihood estimate. That

made the computation possible in the time of the publication of his paper. Then, instead of using the observed covariate vectors, an arbitrary value was permitted. In the simplest case, all covariates may be assumed $\vec{x} = \vec{0}$, which would completely nullify the covariate contribution into the failure rate function, thus enabling us to infer it entirely separately. Cox himself has used \vec{x} being the mean of the covariates of the relevant risk set in order to minimise the variance of the non-parametric estimator he used.

Once the inference is done, the (conditional) survival function can be evaluated at any time instance as

$$R(t|\vec{x}, \vec{\beta}, \lambda_0) = [R_0(t)]^{\exp(\vec{\beta}\vec{x})} = \left[\exp\left(-\int_0^t \lambda_0(y)dy\right) \right]^{\exp(\vec{\beta}\vec{x})}.$$

4.4.4 *Quality Control*

Another important inferential task in reliability theory is connected to quality control [16], [17, Ch. 13]. Imagine that we have a factory producing certain device. Since, again, the factory itself is a device operating in the real world, its actual performance may be influenced by environmental disturbances, and the resulting products may vary in quality. This may also affect the reliability of the products. In order to assure that a certain quality of the products is met, they need to be regularly tested. Let us divide the production into batches, sets of products produced under the same (similar) conditions and in the same time frame. Take a single batch and assume that all the products in this batch share the same failure time distribution. We would then be interested whether this common failure distribution meets the required criteria. Besides the reliability, we might be interested also in other varying quality measures. Answering this question requires us to find a balance among two conflicting demands on the testing procedure. The larger amount of products we subject to testing, the more reliable the answer we will obtain should be. But also, the less products we test, the more of them we can actually monetise, since the tests are often destructive.

A certain quality might be required, say that the mean lifetime is larger than some value or that the geometry is within specified tolerances. Satisfying the requirement may be viewed as a random event, say H as hypothesis, and in practice may be solved by formulating a hypothesis test based on a set of observed failure times. The hypothesis test might result into four different outcomes:

1. H is valid and the test concludes that.
2. H is not valid and the test concludes that.
3. H is valid, but the test concludes otherwise—I. type error.
4. H is not valid, but the test concludes otherwise—II. type error.

So, again, we cannot determine whether H is valid or not for certain based on the data only. There is always a possibility that a test would conclude incorrectly, but the probability that would occur may be controlled. If we were given a fixed set of observation, we are usually able to control just one of the two types of error. In practice we usually choose to control the type I. error by selecting the test significance level accordingly. In the quality testing, the two errors represent the *risks* to the producer (type I., the whole batch would be rejected wrongfully) and to the purchaser (type II., a batch would not meet the requirements although the test concluded that). In the batch testing procedure, we are able to control both error levels by selecting a proper amount of products to test, since for fixed I. type error, the type II. error generally decreases with increasing number of observations. The question that remains is an economical one, what error levels we deem adequate given that lowering them could be expensive.

4.5 Designing Highly Reliable Systems

Once we are able to construct mathematical models for the occurrence of failure, the next logical step is to select the design which best suits our needs. Reliability is a sidekick to performance. Even if one designs a system with the peak performance, it would be of little use if it would stay in a failure state most of the time. Generally, we require the failure probability to be as small as possible. But engineering has its limits and may only take us so far with constructing reliable devices. This section is to introduce techniques which might be used to improve reliability of critical systems other than by improving reliability of its components. It is theoretically possible to construct devices with any required reliability level, but that is usually impossible in practice due to the possible correlations among the failure times (when one failure triggers others) or due to possible failure of switching mechanisms. Nevertheless, the reliability of a system and its lifetime may be increased drastically. The pay-off is the cost, spatial dimension and complexity.

4.5.1 Redundancy Allocation

One idea leading to an increase in reliability is to identify and fortify critical components of the systems [4, Ch. 6], [14, Ch. 9]. If there is a component whose failure will likely lead to the failure of the whole system and whose reliability cannot be sufficiently improved by changing its design and construction, we might consider to introduce additional components which would be able to ease the stress upon this component (share its load) or which could substitute it in the case of a failure (redundant spares). Both of these may simply be just replicas of the original component.

As indicated in the ALT Sect. 4.4.2, higher stress upon a component might lead to its faster deterioration. If we introduce another component which takes on part of the original component's function, we may be able to achieve increased lifetimes of both components due to decreased stress, and, even without decrease in load, increased system lifetime due to the fact that upon the failure of one of these components, the other can still fulfil the functionality by taking the load of the other (if the system fails only when all these components fail).

The dependency of lifetime on stress may be examined, for example, by the proportional hazard model (Sect. 4.4.3) or by an inferred model from accelerated life tests (Sect. 4.4.2) and taken into account in the system model. Due to the introduced redundancy, the component together with the introduced support systems may be modelled as a macro-component with a new TTF given by

$$\text{TTF}_{\text{load share system}} = \max\{\text{TTF}_1, \dots, \text{TTF}_n\}, \quad (4.4)$$

where n is a number of components in this load-share system and TTF_i their respective TTFs (which might be possibly influenced by the failures of other sub-components by the increase of the remaining components failure rate functions).

In contrast to the load-sharing setting, we can also introduce spares, stand-by components which do not operate (and we assume that they also do not deteriorate) until the original component fails, and then they replace its function. This new configuration of N components may again be viewed as a macro-component itself with new TTF

$$\text{TTF}_{\text{system with spares}} = \sum_{i=1}^N \text{TTF}_i, \quad (4.5)$$

where TTF_i denotes the TTF of the i th sub-component.

Figure 4.8 depicts how different scenarios of load sharing influence the system reliability. Stand-by components are not considered there. We consider a system with single component with exponentially distributed lifetime and compare its survival function (the curve "Original") with scenarios where a redundant component is introduced in the system assuming that the load share:

- does not influence components' lifetime, so the overall TTF is given by Eq. (4.4) with the original component TTFs (the curve "Same stress"),
- increases the lifetime of components by decreasing its failure rate and a failure of one component does not influence the failure rate of the other, so the overall TTF is given by Eq. (4.4) with TTFs with decreased failure rate (the curve "Decreased stress"),
- increases the lifetime of components, as in the previous case, but a failure of one of the components increases stress on the other by imposing the original failure rate for the rest of its life (the curve "Change of stress").

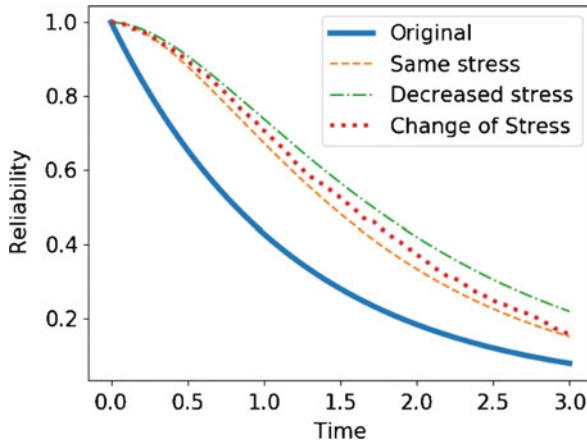


Fig. 4.8 Survival functions for different considered redundant system scenarios. “Original” for a single component system; “Same stress” for when we assume original stress on both components; “Decreased stress” for when we assume decreased stress level regardless the state of the other component; “Change of stress” for when the stress is divided while both components are functioning, but increased once one of the components fails

Notice also, how the scenario in which the stress upon a component changes after the other component’s failure can be bounded between the curves corresponding to situations with the decreased stress and the original level of stress. This is due to the ordering of the respective new TTFs induced by ordering of their failure rates.

Redundancy allocation is often the key to designing highly reliable devices, especially important for missions where failure of the system leads to catastrophic consequences (e.g. life losses or loss of deep space probes). One must compensate the improvements in reliability with an increase of costs (and weight, size, ...).

The task itself leads to an integer optimisation problem—for each of the components, we may allocate an arbitrary number of redundant ones in various schemes (load share and spare). The problem is often solved sequentially with the help of **sensitivity analysis**, which allows us to locate critical parts of the system and assign redundant components to them. Then we may iterate until the desired reliability level is reached.

4.5.2 System Maintenance

Maintenance refers to a set of procedures developed to attend to systems after their deployment. It enables us to drastically prolong system lifetimes and/or keep it operational even after its original lifetime by overhauling the systems and replacing its failed components [12], [14, Ch. 10], [18].

Whatever we build will eventually fail. Nevertheless, we would like to access the service that a device is providing for us for any time period we choose. We can (and do) maximise the lifetime of a device by improving its design, but in order to enable the service to be prolonged up to an arbitrary amount of time, we sometimes have to replace the failed device or its parts with new, functioning ones or perform a repair to make them functional again. Maintenance theory is generally concerned with the overall system operation, including its economical aspects, like the costs and logistics of enabling the repair at all. In this section, we will introduce some basic aspects of the maintenance in the current subsection.

In renewal theory, we are no longer focusing on reliability of a mission—meaning mission is successful if system does not fail until the mission's end. We now admit that a failure may occur and shift our focus to the performance measures instead—i.e. what proportion of time is the system operational (and possibly how that influences other performance measures). In this scenario, we assume that once a system (or its components) fails, a repair process is initiated with a random **time to renewal** denoting its duration, a time span after which the system will usually regain its functionalities. Variations exist when failures are not directly observed, and we need to plan also for inspections of the system or when the repair does not renew the system but only make it *minimally* functional again. The state of the system is again a random process in time with states $\{0, 1\}$ as before, but now we pose no restrictions on monotonicity.

The renewal function $N(t)$ describes the number of repairs in interval $[0, t]$. It depends on the design of the system and the policies for its maintenance and directly influences the economical aspects of the system (i.e. how much will it cost to maintain the operation of the system, or can we supply enough spare parts?). Because the times to failure are random variables, the renewal function will be a random process, and mostly, for the sake of simplicity, we focus on its mean value.

Another important performance measure is the function describing the **availability** of a system—the probability that system is functioning at a specific moment in time. Even though we can replace a failed component, the replacement may not be immediate so the provided service may be unavailable (not functioning) during the time of maintenance. The availability is, again, a time dependent function, which we will denote as $A(t) : T \rightarrow [0, 1]$.

Specific form of $N(t)$, $A(t)$ and $\mathbb{E}\{N(t)\}$ (the mean value of $N(t)$) depends on the qualitative properties of the system, namely on the models of the processes of maintenance and failure inspection. Many real world systems will fall into one of the following categories.

- **Systems with immediate repair**—the maintenance length is negligible so a renewed unit is considered to start working immediately after the last unit failure. The system itself is considered operational at all times ($A \equiv 1$), and we are only interested in $N(t)$.
- **Systems with significant maintenance time**—the maintenance length cannot be neglected, but maintenance still commences right after the failure. The system may not be operational at any time, since it may be undergoing maintenance.

- **Systems with latent failures**—here we can again distinguish based on the negligibility of maintenance length, but the main aspect is that the maintenance does not commence directly at the time of a device failure but only at predefined inspection times. Hence the inspection policy also becomes an aspect of the system design.

Results of the **renewal theory** [12], [18, Ch. 1] directly lead to methods for optimising the overall performance of the service by the means of influencing the maintenance and inspection policies. The competing objectives are the availability of the system (preferred maximal), the overall costs of the maintenance (preferred minimal) and others relevant for the system operation, like the variability of its performance (preferred minimal).

Approximative results can be derived from the limiting properties of $N(t)$ and $A(t)$, namely: $\lim_{t \rightarrow \infty} A(t)$ and $\lim_{t \rightarrow \infty} \frac{N(t)}{t}$. These describe the behaviour of maintained devices at times after the initial fluctuations and have enough descriptive power to reason about long-lasting or permanent systems. For the scenario with significant repair time, but no latent failures, the asymptotic availability is given by

$$\lim_{t \rightarrow \infty} A(t) = \frac{MTTF}{MTTF + MTTR}, \quad (4.6)$$

where $MTTF$ and $MTTR$ represent the **mean TTF** and the **mean time to renewal**, respectively.

But Eq. (4.6) only describes the asymptotic properties, and for bounded system life duration, these might not be reached and therefore be misleading. A careful, usually simulation-based, analysis is necessary in such cases.

4.6 Concluding Remarks

In this chapter, we have attempted to give an overview of the basic notions and problems reliability theory deals with. For the sake of conciseness, we needed to omit many interesting topics and sometimes also proper mathematical rigour. Nevertheless, for those interested, several pointers to further literature were provided in relevant sections.

Regarding the optimal design problem, some aspects of dependency of reliability on design variables has been emphasised:

- Reliability of a system depends on its structure, so the structure itself becomes a design parameter (Sect. 4.3).
- The design may influence the working conditions of the system, the stresses on components and the models, for this may be inferred via accelerated life tests (Sect. 4.4.2).
- “Linear” dependencies on additional variables may be captured by the proportional hazard model (Sect. 4.4.3).

- Reliability of a system may be further improved by adding redundant components and planning for maintenance, which introduces additional design variables (Sect. 4.5).

Reliability theory also suffers from the same drawback like the general uncertainty quantification, as introduced in Chap. 2. Namely, the available information is often too scarce to construct precise stochastic models. In reliability engineering, knowledge often comes in the form of distribution summaries (mean TTF and other moments supplied by the manufacturer) or only via limited amount of samples (as with testing highly reliable components). It has been argued [23, 24] that imprecise probability models are necessary in order to obtain reliable predictions.

The problem of inference with limited assumptions was already tackled also in the treatment of Barlow and Proschan [4], who had derived several inequalities for bounding the survival functions based on combination of quantitative (moments of the lifetime distributions) and qualitative judgements (whether the failure time distribution has increasing or decreasing failure rate). Further extensions to imprecise probability framework have been achieved in the field of robust Bayesian inference [26], in analysing censored datasets via NPI [8], in ALT through imprecise transformation of observations to the base level [27] and more in the field of system reliability where imprecise failure distributions of component lifetimes may be extended to imprecise reliability of some basic systems [19, 25].

References

1. J.D. Andrews, D.R. Prescott, R. Remenyte-PreScott, A systems reliability approach to decision making in autonomous multi-platform systems operating a phased mission, in *2008 Annual Reliability and Maintainability Symposium* (2008), pp. 8–14
2. J.D. Andrews, J. Poole, W.-H. Chen, Fast mission reliability prediction for Unmanned Aerial Vehicles. *Reliab. Eng. Syst. Saf.* **120**, 3–9 (2013)
3. T. Augustin et al. (eds.), *Introduction to Imprecise Probabilities* (Wiley, New York, 2014)
4. R.E. Barlow, F. Proschan, *Mathematical Theory of Reliability/Richard E. Barlow, Frank Proschan, with contributions by Larry C. Hunter* [English] (Wiley, New York, 1967)
5. P. Bessière et al., *Bayesian Programming* (CRC Press, Boca Raton, 2013)
6. G. Casella, R.L. Berger, *Statistical Inference* (Thomson Learning, Pacific Grove, 2002)
7. F.P.A. Coolen, T. Coolen-Maturi, Generalizing the signature to systems with multiple types of components, in *Complex Systems and Dependability*, ed. by W. Zamojski et al. (Springer, Berlin, 2012), pp. 115–130
8. F.P.A. Coolen, K.-J. Yan, Nonparametric predictive inference with right-censored data. *J. Stat. Plan. Inference* **126**, 25–54 (2004)
9. D.R. Cox, Regression models and life-tables. *J. R. Stat. Soc. Series B Methodol.* **34**, 187–220 (1972)
10. J.D. Esary, H. Ziehms, Reliability analysis of phased missions. Tech. rep., Naval postgraduate school, Monterey, California, 1975
11. D.F. Haasl et al., *Fault Tree Handbook* (US Nuclear Regulatory Commission, Washington, 1981)
12. A.K.S. Jardine, *Maintenance, Replacement and Reliability* (Halsted Press, Wiley, New York, 1973)

13. R.S. Kennet, F. Ruggeri, F.W. Faltin (eds.), *Analytic Methods in Systems and Software Testing* (Wiley, New York, 2018)
14. E.E. Lewis, *Introduction to Reliability Engineering* (Wiley, New York, 1995)
15. W.B. Nelson, *Accelerated Testing: Statistical Models, Test Plans, and Data Analysis* (Wiley, New York, 2004)
16. P. O'Connor, *Test Engineering: A Concise Guide to Cost-Effective Design, Development and Manufacture* (Wiley, New York, 2001)
17. P. O'Connor, A. Kleyner, *Practical Reliability Engineering* (Wiley, New York, 2012)
18. S. Osaki (ed.), *Stochastic Models in Reliability and Maintenance* (Springer, Berlin, 2002)
19. E. Patelli et al., Simulation methods for system reliability using the survival signature. *Reliab. Eng. Syst. Saf.* **167**, 327–337 (2017)
20. F.J. Samaniego, *System Signatures and Their Applications in Engineering Reliability* (Springer US, Berlin, 2007)
21. N.D. Singpurwalla, *Reliability and Risk: A Bayesian Perspective* (Wiley, Chichester, 2006)
22. M. Todinov, *Reliability and Risk Models: Setting Reliability Requirements* (Wiley, New York, 2015)
23. L.V. Utkin, F.P.A. Coolen, Imprecise reliability: an introductory overview, in *Computational Intelligence in Reliability Engineering: New Metaheuristics, Neural and Fuzzy Techniques in Reliability*, ed. by G. Levitin (Springer, Berlin, 2007), pp. 261–306
24. L.V. Utkin, S.V. Gurov, New reliability models based on imprecise probabilities, in *Advanced Signal Processing Technology by Soft Computing*, ed. by C. Hsu (World Scientific, River Edge, 2001), pp. 110–139
25. L.V. Utkin, I.O. Kozine, Computing the reliability of complex systems, in *Proc 2nd International Symposium on Imprecise Probabilities and Their Applications* (Shaker Publishing, Maastricht, 2001), pp. 324–331
26. G. Walter, L.J.M. Aslett, F.P.A. Coolen, Bayesian nonparametric system reliability using sets of priors. *Int. J. Approx. Reason.* **80**, 67–88 (2017)
27. Y.-C. Yin, F.P.A. Coolen, T. Coolen-Maturi, An imprecise statistical method for accelerated life testing using the power-Weibull model. *Reliab. Eng. Syst. Saf.* **167**, 158–167 (2017)

Chapter 5

An Introduction to Imprecise Markov Chains



Thomas Krak

Abstract Stochastic processes in general provide a popular framework for modelling uncertainty about the evolution of dynamical systems. The theory of Markov chains uses a number of crucial assumptions about the (in)dependence of such a process on its history that make their analysis tractable. In practice however, the parameters of a Markov chain may not be known exactly, or there may exist doubt as to the applicability of these assumptions to the system under study. This chapter presents an introduction to imprecise Markov chains, which are a robust generalisation of these models that may be used when parameters are not known exactly or when such assumptions could be violated. Their treatment is grounded in the theory of imprecise probabilities. The generalised model can be interpreted as a set of (traditional) stochastic processes, which may or may not be Markovian and which may have different and varying parameter values. Inferences are then performed to ensure robustness with respect to variations within this set. This chapter assumes no advanced familiarity with Markov chains or imprecise probability theory. It aims to develop an intuitive and graphical understanding of (imprecise) Markov chains in discrete and in continuous time.

Keywords Imprecise probabilities · Model uncertainty · Stochastic processes · Imprecise Markov chains

5.1 Introduction

In many areas of science and engineering, we are interested in modelling uncertainty about the behaviour of dynamical systems, that is, systems whose state changes as time passes. For instance, we may want to model the evolution of the spatial trajectories of a system in motion; or the performance and reliability of a complex

T. Krak (✉)
IDLab, Ghent University, Ghent, Belgium
e-mail: thomas.krak@ugent.be

composite system as its components wear out, break down and get replaced; or the spread of pathogens through a population; or the evolution of stock prices—and so on and so forth.

What all these systems have in common is that there is a *dynamic* component to their description—they change *over time*—and they are, in a sense, hard to describe *exactly*. For instance, this difficulty may arise because their behaviour depends on unknown external influences or because the system cannot reasonably be described at a sufficiently detailed level. Thus, there arises an uncertainty about how exactly the system will evolve over time, even if one can model how it will ‘roughly’ behave. Regardless of the interpretation that we want to assign to this uncertainty, such systems are modelled using *stochastic processes*. A stochastic process, then, is a probabilistic description of the system under study. In this sense, it provides a formal and integrated description of the system dynamics and the probabilistic uncertainty of its evolution.

On the other hand, we might also be uncertain about whether such a model is ‘correct’. For instance, we might not know exactly the numerical values that the parameters of our model should take. Similarly, we might be aware that our modelling assumptions lead to simplifications that are not necessarily warranted, which introduces uncertainty about the accuracy or applicability of any assessments made on the basis of these models. It is therefore of interest to robustify our models also against these kinds of ‘meta’, or ‘higher-order’, uncertainties.

In this chapter, we consider stochastic processes for which this higher-order uncertainty is modelled using the theory of imprecise probabilities (IP). For an extended introduction to IP, we refer the reader back to Chap. 2. We constrain ourselves to briefly recalling that such imprecise probabilistic models can be interpreted as representing a *set* of traditional probabilistic models. So, in our current setting, we will be considering *sets* of stochastic processes. From an inference point of view, the aim is then to compute inferences which are robust with respect to variations within such a set. We recall from Chap. 2 that these robust inferences are captured in general by the *lower* and *upper* expectations with respect to the elements of the set that we are considering.

Our aim with the present chapter is to provide an extensive but intuitive introduction to the theory of imprecise stochastic processes and of imprecise Markov chains in particular. To this end, we will intentionally focus on the different representations of these processes. We will show how each of the different ways of looking at these models provides its own way of deriving useful properties and highlights different intuitive ways of reasoning about them. Important results and properties are stated, but we have made an effort to keep the discussion intuitive. We try to prevent technicalities and do not provide extended proofs; instead, we will provide pointers to the literature that the interested reader might pursue herself.

The remainder of this chapter is organised as follows. We start the discussion by giving a quick introduction to stochastic processes in Sect. 5.2. The first part basically uses the measure-theoretic approach (albeit in a rather simplified sense) to pin down some first concepts and notation. We then go on to present three different and graphical representations of stochastic processes, which can be used

when the time-dimension is discrete. Specifically, we cover the representation using probability trees, in Sect. 5.2.1; using Bayesian networks, in Sect. 5.2.2 and using transition graphs, in Sect. 5.2.3.

Once we have developed these different ways of reasoning about discrete-time processes, we generalise the discussion to *imprecise* discrete-time processes in Sect. 5.3. We use the previously developed graphical notions to provide intuition about how to reason and compute inferences using these models. The treatment of (imprecise) continuous-time processes is largely postponed until Sect. 5.4. Here the graphical and intuitive representations largely break down, but we can then use the previously developed understanding of the discrete-time case to reason about these models. To keep the main text as readable as possible, the discussion of the literature on which the material in this chapter is based is deferred to Sect. 5.5.

5.2 (Precise) Stochastic Processes

We will start the exposition around stochastic processes in a relatively general and abstract sense but will quickly make things more specific. Throughout the remainder of this chapter, we will consider some fixed abstract *state-space* \mathcal{X} . A *state* is an element $x \in \mathcal{X}$ and represents uniquely the relevant information about the underlying system that we are interested in modelling. So as not to complicate matters, we will assume throughout that \mathcal{X} is finite, so that we can identify it without loss of generality as the set $\mathcal{X} = \{1, \dots, k\} \subset \mathbb{N}$. Note that here and in what follows, we denote with \mathbb{N} the natural numbers and will write $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ when we include zero. Furthermore, the real numbers are written \mathbb{R} , the non-negative reals are $\mathbb{R}_{\geq 0}$ and the positive reals are $\mathbb{R}_{> 0}$.

Because we are interested in modelling a system whose state $x \in \mathcal{X}$ changes over time, we next identify some *time-dimension* \mathbb{T} . A crucial choice to be made later on is whether we are considering processes in discrete-time, in which case we identify $\mathbb{T} = \mathbb{N}_0$, or processes in continuous-time, in which case $\mathbb{T} = \mathbb{R}_{\geq 0}$. For now we simply keep the discussion general without making this identification.

With the state-space and time-dimension in place, it now makes sense to talk about the *realisation* of some (yet to be identified) stochastic process. Such a realisation is also called a *sample path*, and it is a function $\omega : \mathbb{T} \rightarrow \mathcal{X}$. So, this ω describes for each point in time $t \in \mathbb{T}$ the state $\omega(t) \in \mathcal{X}$ that the system was in at that time. We collect in the set Ω all these sample paths. For technical reasons, it is sometimes required to restrict attention to paths that satisfy sufficient smoothness conditions; for instance, when $\mathbb{T} = \mathbb{R}_{\geq 0}$, it is common practice to let Ω only contain càdlàg functions, that is, paths $\omega(t)$ that are right-continuous and whose left-sided limits exist everywhere.

This set Ω thus contains all possible ways in which the system might behave over time; it can therefore be considered an *outcome space* of a stochastic model. Formally, we will consider some abstract underlying probability space (Ω, \mathcal{F}, P) , where \mathcal{F} is some appropriate σ -algebra on Ω and where P is a probability measure

on (Ω, \mathcal{F}) . Given this probability space, we can finally formalise the notion of a *stochastic process* as a collection $\{X_t\}_{t \in \mathbb{T}}$ of random variables associated to this probability space. We will here slightly restrict our definition to the following specific stochastic process:

Definition 5.1 (Stochastic process) Fix a time-dimension \mathbb{T} and consider a probability space (Ω, \mathcal{F}, P) . Then (the corresponding) stochastic process is the collection $\{X_t\}_{t \in \mathbb{T}}$ of random variables $X_t : \Omega \rightarrow \mathcal{X} : \omega \mapsto \omega(t)$, $t \in \mathbb{T}$, on this space.

Corollary 5.1 Fix a time-dimension \mathbb{T} ; consider a probability space (Ω, \mathcal{F}, P) ; and let $\{X_t\}_{t \in \mathbb{T}}$ be the corresponding stochastic process. Then for all $t \in \mathbb{T}$ and $x \in \mathcal{X}$, it holds that $\Pr(X_t = x) = P(\{\omega \in \Omega : \omega(t) = x\})$.

Proof Fix $t \in \mathbb{T}$, and recall the definition of a random variable: for all $x \in \mathcal{X}$, the probability $\Pr(X_t = x)$ of X_t taking the value x is equal to $P(X_t^{-1}(x))$, the measure of its preimage in Ω . Since $X_t(\omega) = \omega(t)$, we have $X_t^{-1}(x) = \{\omega \in \Omega : \omega(t) = x\}$. \square

The above is a formal way of saying that, and how, these random variables $\{X_t\}_{t \in \mathbb{T}}$ are associated to the given probability space. In words, for some fixed time $t \in \mathbb{T}$, X_t is a random variable that takes on a value $x \in \mathcal{X}$ with probability equal to the measure of the set of paths along which the state at time t is x . Conversely, if we fix the outcome $\omega \in \Omega$, then the collection $\{X_t\}_{t \in \mathbb{T}}$ can be considered a deterministic process, and $X_t(\omega) = \omega(t)$ for all $t \in \mathbb{T}$.

Note, therefore, that all the quantitative information about the probability of the process taking on certain values at given points in time are completely determined by the measure P . It is therefore also intuitive to instead consider this measure P to be ‘the stochastic process’, although this is technically an abuse of terminology. This is because, for a given probability space (Ω, \mathcal{F}, P) , it is possible to define many different stochastic processes; any \mathbb{T} -indexed collection of random variables on this space satisfies the general definition. However, in a sense, the stochastic process in Definition 5.1 can be viewed as the ‘canonical’ stochastic process corresponding to the given probability space, since it specifically and exactly represents the uncertainty about which states might be obtained at different points in time. We will therefore, and for notational convenience, often refer to the measure P and its corresponding stochastic process $\{X_t\}_{t \in \mathbb{T}}$ interchangeably and without confusion.

Next, it will be convenient to have a standardised notation to index a subset of the random variables of a stochastic process. To this end, for any finite sequence of time points $\mathbf{t} = t_1, \dots, t_n$ in \mathbb{T} , with $n \in \mathbb{N}$, we will write $X_{\mathbf{t}} = X_{t_1}, \dots, X_{t_n}$. Typically, these sequences will be taken to be ordered, so that $t_1 < \dots < t_n$. Note that each of the random variables X_{t_i} , $i = 1, \dots, n$ takes values in \mathcal{X} . Hence, the sequence $X_{\mathbf{t}}$ takes values (jointly) in $\mathcal{X}^n = \times_{i=1}^n \mathcal{X}$. An element of this joint state-space is thus a vector $(x_1, \dots, x_n) \in \mathcal{X}^n$. When we are explicitly talking about a sequence \mathbf{t} of n time points, we will also write $x_{\mathbf{t}}$ to denote a generic element of \mathcal{X}^n .

In what follows, we will be interested in computing the expectation of some real-valued function, whose value depends on the specific realisation of the stochastic process. To prevent technical difficulties, we will assume that this function only depends on a finite number of time points; without loss of generality, we can then assume that it is a map $f : \mathcal{X}^n \rightarrow \mathbb{R}$, with $n \in \mathbb{N}$, whose value depends on the n random variables $X_{\mathbf{t}}$, with $\mathbf{t} = t_1, \dots, t_n$ in \mathbb{T} . We collect in the set $\mathcal{L}(\mathcal{X}^n)$ all such real-valued functions on \mathcal{X}^n . The expected value of any such $f \in \mathcal{L}(\mathcal{X}^n)$ on the n time points \mathbf{t} is defined as

$$\mathbb{E}_P[f(X_{\mathbf{t}})] = \sum_{x_{\mathbf{t}} \in \mathcal{X}^n} f(x_{\mathbf{t}})P(X_{\mathbf{t}} = x_{\mathbf{t}}), \quad (5.1)$$

where we have implicitly introduced the intuitive notation for the set

$$(X_{\mathbf{t}} = x_{\mathbf{t}}) = \left\{ \omega \in \Omega : (\forall i \in \{1, \dots, n\} : \omega(t_i) = x_{t_i}) \right\}.$$

In Eq. (5.1), we use the subscript P for the expectation operator \mathbb{E}_P to make explicit that it is taken with respect to the measure P ; this will be notationally convenient further on.

We finish this first introduction by recalling the notion of conditional probabilities and conditional expectations. For any two finite sequences of time points \mathbf{t} and \mathbf{s} in \mathbb{T} , the *conditional probability* of $X_{\mathbf{t}}$, given $X_{\mathbf{s}}$, is derived using *Bayes' rule*:

$$P(X_{\mathbf{t}} | X_{\mathbf{s}}) = \frac{P(X_{\mathbf{s}}, X_{\mathbf{t}})}{P(X_{\mathbf{s}})},$$

whenever $P(X_{\mathbf{s}})$ is strictly positive. The necessity of the final condition is obvious; it leads to a division by zero whenever it does not hold.

Using this notion of conditional probability, we can define conditional expectations analogously. Suppose the sequences \mathbf{s} and \mathbf{t} are of length $n, m \in \mathbb{N}$, respectively. Then for any $f \in \mathcal{L}(\mathcal{X}^{n+m})$ on $X_{\mathbf{s}}, X_{\mathbf{t}}$ we define, for all $x_{\mathbf{s}} \in \mathcal{X}^n$,

$$\mathbb{E}_P[f(X_{\mathbf{s}}, X_{\mathbf{t}}) | X_{\mathbf{s}} = x_{\mathbf{s}}] = \sum_{x_{\mathbf{t}} \in \mathcal{X}^m} f(x_{\mathbf{s}}, x_{\mathbf{t}})P(X_{\mathbf{t}} = x_{\mathbf{t}} | X_{\mathbf{s}} = x_{\mathbf{s}}).$$

5.2.1 Probability Trees

The preceding discussion introduced stochastic processes in a very general, but rather abstract sense. We will build further intuition by next offering a different view and representation, by means of *probability trees*. In the remainder of this section, unless otherwise specified, we will focus on discrete-time stochastic processes, whence we identify $\mathbb{T} = \mathbb{N}_0$.

We next need some notation and definitions for ‘partial paths’, which in this setting are also called *situations*. As before, a (full) path is a map $\omega : \mathbb{N}_0 \rightarrow \mathcal{X}$. In contrast, a *situation* is defined as a (finite length) *prefix* of such a path. In other words, a situation is an element of a set \mathcal{X}^n , for some $n \in \mathbb{N}$. If $w \in \mathcal{X}^n$, $n \in \mathbb{N}$, is a situation, we write w_i for its $(i + 1)$ -th coordinate, $i \in \{0, \dots, n - 1\}$, and we say that its *length* is $|w| = n$. Note that the indexing over the coordinates is taken to start from zero rather than one—this is done for notational consistency with paths ω . Since we will need to refer to it so often, we introduce the shorthand notation w_\top for the *last* element of w ; so if w has length n , then $w_\top = w_{n-1}$. The set of all non-empty situations is $\mathcal{X}^* = \cup_{n \in \mathbb{N}} \mathcal{X}^n$, and we define $\mathcal{X}_\square^* = \{\square\} \cup \mathcal{X}^*$, where we add the *empty situation* denoted by \square .

As a final point in this notational digression, for any $s, t \in \mathbb{N}_0$ such that $s \leq t$, we will introduce the shorthand notation $s : t$ to denote the sequence of time points s, \dots, t . Using our previously introduced notation, we can then write $X_{s:t}$ for the random variables at these time points. Furthermore, for any $n \in \mathbb{N}_0$ and any situation $w \in \mathcal{X}^{n+1}$, we can then use the previously introduced notation to write $X_{0:n} = w$; this is understood to mean that the random variables at time points $0, \dots, n$ obtained the states corresponding to the situation w .

We endow the set \mathcal{X}_\square^* with the *prefix order*, denoted $<$, which is a partial order such that $\square < v$ for all $v \in \mathcal{X}^*$ and for all $v, w \in \mathcal{X}^*$ with lengths $n = |v|$ and $m = |w|$, it holds that $v < w$ if and only if $n < m$ and $v_i = w_i$ for all $i \in \{0, \dots, n - 1\}$. This is just a rigorous but somewhat obfuscated way of saying that $v < w$ if ‘ v is the beginning of w ’ or ‘ w is what you can get if v happens first, and then some other things happen’ or, indeed, ‘ v is a prefix of w ’.

The important thing to notice is that the ordered set $(\mathcal{X}_\square^*, <)$ induces a graphical tree structure, with all the situations as its vertices. This tree is what is known as the *event tree*. It has \square as its root, and, for all $v, w \in \mathcal{X}_\square^*$, w is a descendant of v exactly if $v < w$. An example of such a tree is shown in Fig. 5.1, which (partially) shows the event tree corresponding to a binary state-space $\mathcal{X} = \{a, b\}$.

Such an event tree can be turned into an intuitive representation of a stochastic process by augmenting it into a *probability tree*. This is done by assigning to each situation $w \in \mathcal{X}_\square^*$ in the tree a *local model* p_w , which is a probability mass function on \mathcal{X} ; that is, it is a map $p_w : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ such that $\sum_{x \in \mathcal{X}} p_w(x) = 1$. An example of this is again illustrated in Fig. 5.1.

Definition 5.2 (Probability tree) A probability tree is a tuple $(\mathcal{X}_\square^*, <, p_{(\cdot)})$, where \mathcal{X}_\square^* is the set of all situations, $<$ is the prefix order on \mathcal{X}_\square^* and $p_{(\cdot)} : \mathcal{X}_\square^* \times \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ represents all local models, so that $\sum_{x \in \mathcal{X}} p_w(x) = 1$ for all $w \in \mathcal{X}_\square^*$.

The mechanism by which a stochastic process obtains a certain realisation $\omega \in \Omega$ can now be interpreted as performing a weighted, random walk along this probability tree, starting from \square . Following the tree in Fig. 5.1, this is done as follows: from \square , we transition either to a , with probability $p_\square(a)$, or to b , with probability $p_\square(b)$. Suppose we transition to a . From this new situation, the next step will take us either to aa , with probability $p_a(a)$, or to ab , with probability $p_a(b)$.

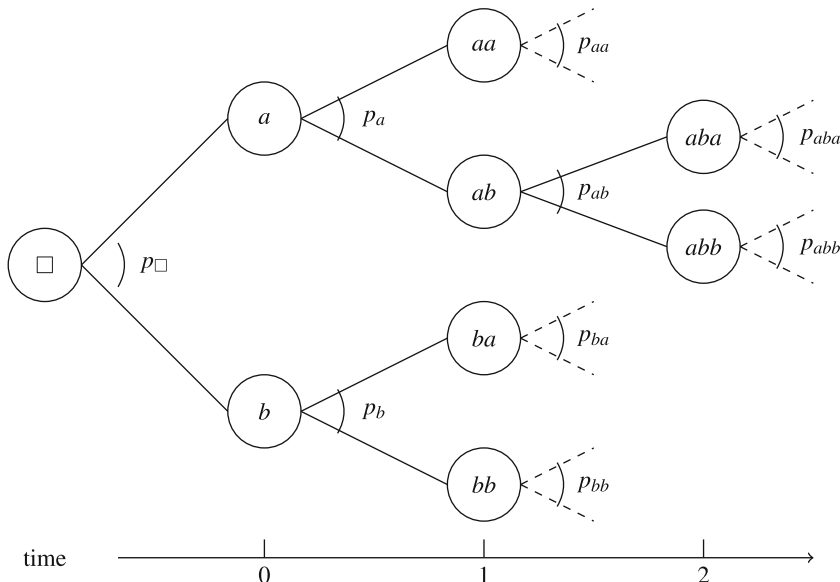


Fig. 5.1 A (partial) event tree for a binary state-space $\mathcal{X} = \{a, b\}$. The vertices are situations, i.e. elements of \mathcal{X}_{\square}^* , and the edges are induced by the prefix order \prec . Dashed lines represent branches that are not shown in the figure. The tree has been augmented to a probability tree, by assigning to each $w \in \mathcal{X}^*$ a local model p_w . A time axis represents at which point in time the situations can occur

Proceeding in this fashion, an infinite random walk along this tree generates a full path $\omega : \mathbb{N}_0 \rightarrow \mathcal{X}$, where, for all $t \in \mathbb{N}_0$, the state $\omega(t)$ represents the (randomly chosen) branch that we took along the tree at the $(t + 1)$ -th step.

This ‘path construction’ view allows us also to connect back to the measure-theoretic definition that we encountered earlier. To obtain this correspondence in one direction, fix a probability tree $(\mathcal{X}_{\square}^*, \prec, p_{(\cdot)})$ and let (Ω, \mathcal{F}) be an appropriate measurable space of discrete-time sample paths, on which we will aim to construct the measure P quantifying, in the measure-theoretic sense, the uncertainty of the corresponding stochastic process $\{X_t\}_{t \in \mathbb{N}_0}$ on the resulting probability space.

We now reason intuitively by using the ‘random walk’ along the probability tree. Starting from \square , we transition to a first situation $x \in \mathcal{X}$ with probability $p_{\square}(x)$. From there, we could then perform the entire infinite random walk to generate the remainder of the path. So, a different way of saying this is that, of all the random paths $\omega \in \Omega$ that could be generated, a fraction of $p_{\square}(x)$ of them will start with $\omega(0) = x$. Using also the interpretation given by Corollary 5.1, it therefore makes sense to define the *first-step marginal measure* $P^*(X_0 = x) \doteq p_{\square}(x)$ for all $x \in \mathcal{X}$.

Let us now consider the next step, and assume the first step down the tree resulted in a situation $x \in \mathcal{X}$. Then, with probability $p_x(y)$, $y \in \mathcal{X}$, the next situation will be xy . In terms of paths that could be generated, a fraction of $p_x(y)$ of the paths that

satisfy $\omega(0) = x$ will furthermore satisfy $\omega(1) = y$. Therefore, we define for the *second*-step marginal measure $P^*(X_0 = x, X_1 = y) \doteq p_{\square}(x)p_x(y)$.

Proceeding in this manner, for every situation $w \in \mathcal{X}^*$ with length $n + 1$, $n \in \mathbb{N}_0$, we can compute the $(n + 1)$ -th step marginal measure as

$$P^*(X_{0:n} = w) \doteq p_{\square}(w_0) \prod_{i=1}^n p_{w_0 \dots w_{i-1}}(w_i),$$

or in words, by multiplying all probabilities given by the local models of the situations encountered on the path from the root of the tree, down to the situation w .

A fundamental result in the measure-theoretic treatment of stochastic processes (known as the *Kolmogorov extension theorem*) states that the collection of all these n -th step marginal measures P^* induces (‘coherently’) a probability measure P on (Ω, \mathcal{F}) . Specifically, the finite n -th step marginals of P will correspond exactly to these n -th step marginal measures that we constructed from the probability tree. This establishes the connection between probability trees and discrete-time measure-theoretic stochastic processes, in that the latter can be constructed from the former.

For the other direction, so, to construct a probability tree from a given probability space (Ω, \mathcal{F}, P) , we start with an event tree $(\mathcal{X}_{\square}^*, <)$ and aim to construct the local models $p_{(\cdot)}$. Using the intuitive interpretation offered by Corollary 5.1, we start by setting $p_{\square}(x) = P(X_0 = x)$ for all $x \in \mathcal{X}$. For all other situations $w \in \mathcal{X}^*$ with length $n + 1$, $n \in \mathbb{N}_0$, the local model p_w is defined as the conditional measure constructed from Bayes’ rule, i.e. for all $x \in \mathcal{X}$,

$$p_w(x) = P(X_{n+1} = x \mid X_{0:n} = w) = \frac{P(X_{0:n} = w, X_{n+1} = x)}{P(X_{0:n} = w)}. \quad (5.2)$$

This also establishes the connection in the other direction. It can be verified that, by now constructing from this probability tree a measure P^* , say, in the manner described above, we obtain again $P^* = P$; so, we conclude that this yields a one-to-one correspondence between probability trees and measure-theoretic stochastic processes.

It should be noted that the second direction in the preceding discussion has one (rather large) caveat: it does not work when there are partial paths that have zero probability to occur. This is because then Bayes’ rule cannot define the conditional measure required to construct the local model for the situation corresponding to that partial path, since it would result in a division by zero.

To summarise, we can conclude that there is indeed a correspondence between the two representations that we have seen so far (up to some technical difficulties surrounding probabilities that are zero). We have seen that the graphical tree structure allows us to reason intuitively about how a stochastic process generates a sample path, by ‘walking’ from the root of the tree down its branches. As we will discuss next, we can also use this structure to ‘reason backwards’: from vertices

deep down in the tree back to the root. We will see that this allows one to intuitively derive *computational methods* for working with stochastic processes.

So, fix $n \in \mathbb{N}_0$, and let $f \in \mathcal{L}(\mathcal{X}^{n+1})$ be a real-valued function for which we aim to compute the expected value with respect to the random variables $X_{0:n}$ at the time points $0, \dots, n \in \mathbb{N}_0$. Note that it suffices to consider this case, in the sense that any function defined on a subset of the variables $X_{0:n}$, can always be trivially extended to a function on all of them. Now first notice the following. For any situation $w \in \mathcal{X}^*$ with length $|w| = n + 1$, the value of f in w is easy to compute; it is simply $f(w)$. Hence in particular, the *expected value* of f , in w , is simply

$$\mathbb{E}[f(X_{0:n}) \mid X_{0:n} = w] = f(w).$$

Recall that the situation w represents a node in the event tree. We will now ‘pull back’ the above expected value, to the time point $n - 1$. Consider therefore the parent situation of w in the probability tree; we will compute the expected value of f in this parent situation.

This parent is a situation v of length $|v| = |w| - 1 = n$, which entirely coincides with w : $v_i = w_i$ for all $i = 0, \dots, n - 1$. Associated to v is the local probability model p_v which, as we have discussed above, represents the probability with which a random walk along the tree travels through the various children of v . In particular, such a random walk goes through the situation w , with probability $p_v(w_\top)$. Therefore, the contribution of the expected value in w , to the expected value in v , is the expected value in w weighted by $p_v(w_\top)$. Since this holds for all children of v , we can write

$$\mathbb{E}[f(X_{0:n}) \mid X_{0:(n-1)} = v] = \sum_{x \in \mathcal{X}} p_v(x) \mathbb{E}[f(X_{0:n}) \mid X_{0:(n-1)} = v, X_n = x].$$

This ‘pullback’ operation is graphically illustrated in Fig. 5.2.

Now, observe that the above conditional expectation of f in v is itself a real-valued function in $\mathcal{L}(\mathcal{X}^n)$. Its value is determined by the states at times $0, \dots, n - 1$. We can therefore repeat the above argument; we pull back to the parent of v , then to the parent of *that* situation and so on. Eventually, the parent that we are considering is the empty situation \square ; we then finish by computing

$$\mathbb{E}[f(X_{0:n})] = \sum_{x \in \mathcal{X}} p_\square(x) \mathbb{E}[f(X_{0:n}) \mid X_0 = x],$$

which is exactly the expected value of f that we started out wanting to compute.

This method to compute the expected value of a function by ‘pulling back’ the ‘local’, or conditional, expected values, uses the interpretation of a stochastic process as a probability tree. The method relies on a property that is called the *law of iterated expectation*, or alternatively the *law of total probability*. It can be

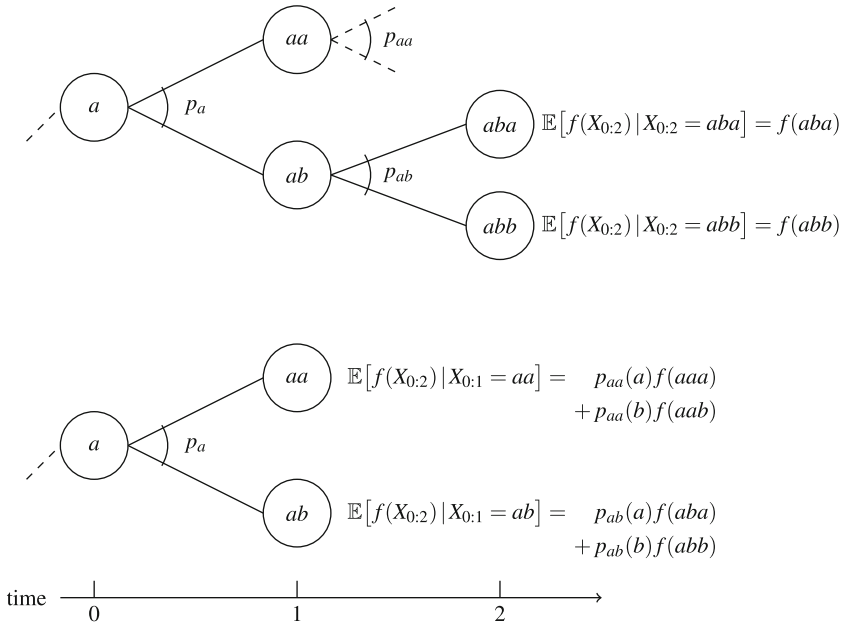


Fig. 5.2 Graphical illustration of ‘pulling back’ the expected value of a function f on $X_{0:2}$, in a probability tree on a binary state-space $\mathcal{X} = \{a, b\}$. Top: the function f is entirely determined by the situations of length 3, i.e. the expected value of the function in those situations is simply the value of the function evaluated in that situation. Bottom: the result after ‘pulling back’ the expectations by one step. The resulting conditional expectation is a function whose value is entirely determined by the situations of length 2. The values are the weighted average of the expectations in the child nodes, weighted by the local models $p_{(\cdot)}$

stated formally in the measure-theoretic context, where it is also easily stated for *continuous-time* stochastic processes.

Theorem 5.1 Fix a time-dimension $\mathbb{T} \in \{\mathbb{N}_0, \mathbb{R}_{\geq 0}\}$, and let $\{X_t\}_{t \in \mathbb{T}}$ be a stochastic process on (Ω, \mathcal{F}, P) . Choose any three ordered sequences $\mathbf{s} = s_1, \dots, s_n$; $\mathbf{t} = t_1, \dots, t_m$ and $\mathbf{u} = u_1, \dots, u_\ell$ in \mathbb{T} , with $n, m, \ell \in \mathbb{N}$ such that $s_n < t_1$ and $t_m < u_1$. Then for any real-valued function $f \in \mathcal{L}(\mathcal{X}^{n+m+\ell})$ on $X_{\mathbf{s}}, X_{\mathbf{t}}, X_{\mathbf{u}}$, it holds that

$$\mathbb{E}[f(X_{\mathbf{s}}, X_{\mathbf{t}}, X_{\mathbf{u}}) | X_{\mathbf{s}}] = \mathbb{E}\left[\mathbb{E}[f(X_{\mathbf{s}}, X_{\mathbf{t}}, X_{\mathbf{u}}) | X_{\mathbf{s}}, X_{\mathbf{t}}] \Big| X_{\mathbf{s}}\right],$$

whenever $P(X_{\mathbf{s}})$ and $P(X_{\mathbf{s}}, X_{\mathbf{t}})$ are everywhere strictly positive.

In this result, the final constraint is required to ensure that the conditional expectations are all well-defined in the measure-theoretic sense. This point did not arise in the discussion using probability trees, because there the local (conditional) models are always properly defined by the model specification.

Having discussed how to interpret probability trees and how to use them to reason about the computation of expected values, we now move on to a discussion of their structural properties. Note that the specification of a probability tree is still relatively complicated. This is not really due to the structure of the tree; the situations \mathcal{X}_\square^* and prefix order $<$ carry enough information to construct the tree up to any desired level, and their mathematical specification is straightforward. However, in order to specify all the local models $p_{(\cdot)}$, we need to provide an infinite number of probability mass functions on \mathcal{X} —one for each situation $w \in \mathcal{X}_\square^*$. This is why one often restricts attention to simpler models, where one needs fewer, and often only finitely many, local models.

These simplifications can be seen as a matter of degree. At the one extreme, we have the general definition that we used above, where each situation $w \in \mathcal{X}_\square^*$ has a local model p_w . This leads to a lot of possible structure but is hard to specify. At the other extreme is the *independent and identically distributed* (i.i.d.) process; this is when we only have a single probability mass function p , and we set $p_w = p$ for all $w \in \mathcal{X}_\square^*$. For such a process, no matter what situation we are in, the next branch will always be chosen according to p . This process is easy to specify, but it does not yield a lot of structure that can capture the dynamics of the underlying system that we are trying to model.

A useful step up from the i.i.d. process is reached by the popular class of models known as *homogeneous Markov chains*. For a homogeneous Markov chain, the local model *only* depends on the last step of the corresponding situation, and not on what happened before that:

Definition 5.3 (Homogeneous Markov chain as probability tree) A probability tree $(\mathcal{X}_\square^*, <, p_{(\cdot)})$ is called a homogeneous Markov chain if $p_v = p_w$ for all situations $v, w \in \mathcal{X}^*$ such that $v_\top = w_\top$.

Corollary 5.2 Let $(\mathcal{X}_\square^*, <, p_{(\cdot)})$ be a homogeneous Markov chain. Then $p_w = p_x$ for all $x \in \mathcal{X}$ and all $w \in \mathcal{X}^*$ such that $w_\top = x$.

Proof Trivial from Definition 5.3 and the fact that all $x \in \mathcal{X}$ are also situations. □

An example for the binary state-space $\mathcal{X} = \{a, b\}$ is shown in Fig. 5.3. Additional degrees of freedom can be introduced back into this model by also letting the local models depend on the corresponding depth of the tree. The dynamics can then depend on the point in time, but not on the *specific* history up to that time. This yields the more general definition of a (non-homogeneous) *Markov chain*:

Definition 5.4 (Markov chain as probability tree) A probability tree $(\mathcal{X}_\square^*, <, p_{(\cdot)})$ is called a Markov chain if $p_v = p_w$ for all situations $v, w \in \mathcal{X}^*$ for which $|v| = |w|$ and $v_\top = w_\top$.

An example for the binary state-space $\mathcal{X} = \{a, b\}$ is shown in Fig. 5.4. It can be verified that a homogeneous Markov chain is a Markov chain, but not—in general—the other way around. Note that, in contrast to homogeneous Markov chains where we only needed to specify local models p_x for all $x \in \mathcal{X}$, we now need different

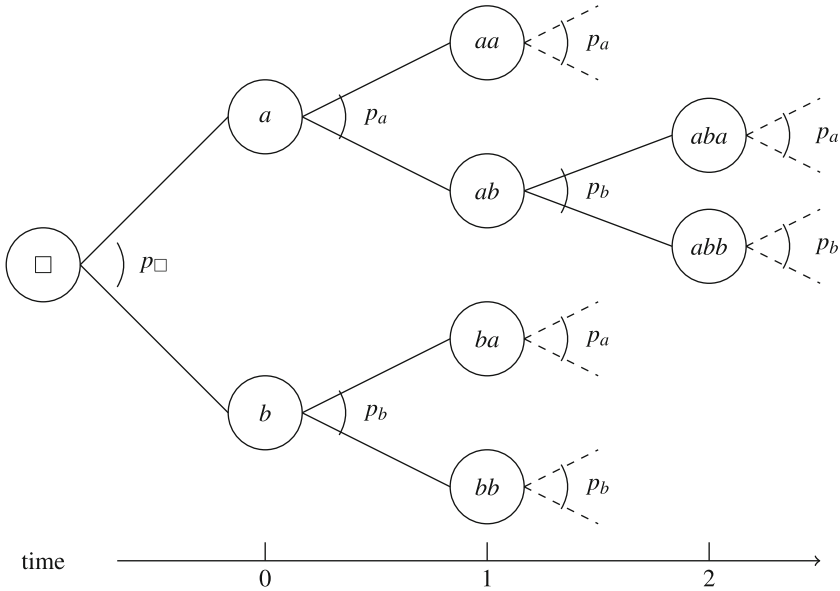


Fig. 5.3 A homogeneous Markov chain, represented as a probability tree

local models for each level of the tree. So, we are now back to needing an infinite number of local models in order to fully describe such a model.

These definitions of (homogeneous) Markov chains can also be conveniently translated back to the measure-theoretic context. We here give the general definition, for an arbitrary time-dimension (so, either $\mathbb{T} = \mathbb{N}_0$ or $\mathbb{T} = \mathbb{R}_{\geq 0}$) and multiple steps into the future:

Definition 5.5 (Markov chain as probability measure) A stochastic process $\{X_t\}_{t \in \mathbb{T}}$ on (Ω, \mathcal{F}, P) is called a Markov chain if for all $s_1, \dots, s_n, t \in \mathbb{T}, n \in \mathbb{N}$, such that $s_1 < \dots < s_n < t$, it holds that $P(X_t | X_{s_1}, \dots, X_{s_n}) = P(X_t | X_{s_n})$. A stochastic process that is a Markov chain is said to have the Markov property.

Similarly, the notion of homogeneity can be defined measure-theoretically and for an arbitrary time-dimension:

Definition 5.6 (Homogeneous Markov chain as probability measure) A stochastic process $\{X_t\}_{t \in \mathbb{T}}$ on (Ω, \mathcal{F}, P) is called a homogeneous Markov chain if it is a Markov chain, and if additionally, for all $s, t \in \mathbb{T}$ such that $s < t$, it holds that $P(X_t | X_s) = P(X_{t-s} | X_0)$.

We leave it as an exercise to verify that, when $\mathbb{T} = \mathbb{N}_0$, Definitions 5.5 and 5.6 correspond to what we would expect from Definitions 5.4 and 5.3, respectively.

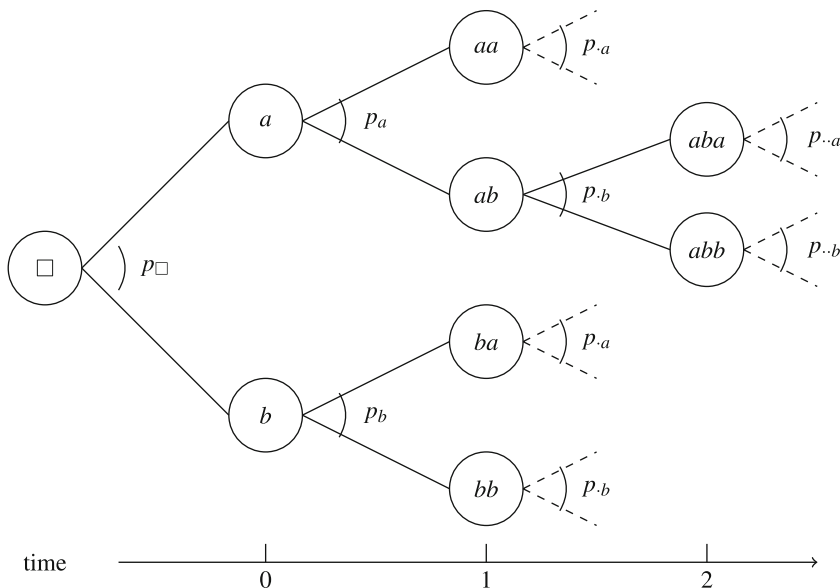


Fig. 5.4 A (non-homogeneous) Markov chain, represented as a probability tree as above

5.2.2 Bayesian Networks

We now move on to a different graphical representation of stochastic processes that is useful for Markov chains in particular: *Bayesian networks* (BNs), a specific type of probabilistic graphical model. While the graphical structure of probability trees in Sect. 5.2.1 emphasised the partial paths in the realisation of a stochastic process, the BN representation emphasises the individual random variables X_t .

The BN representation of a discrete-time Markov chain $\{X_t\}_{t \in \mathbb{N}_0}$ is given in Fig. 5.5. The structure is a directed acyclic graph, with one node associated to each random variable X_t and arcs representing the dependence of the receiving node’s random variable’s distribution, on the originating node’s random variable’s value. Due to the Markov property (c.f. Definition 5.5), each random variable X_n , $n \in \mathbb{N}$, is only (‘directly’) dependent on X_{n-1} , the value of the random variable immediately before it. The initial variable X_0 is somewhat of a special case, since it does not depend on any other variables; there are no time points preceding it. Due to these properties, the graphical structure is that of a chain; this may go some way in explaining the name ‘Markov chain’. In the remainder of this section, we will refer to both a node in the BN and to its random variable, using the notation X_t .

It should be emphasised that the graphical structure is not saying that only nodes which are adjacent in the BN can influence each other. The formal interpretation is as follows: for any node X_n , $n \in \mathbb{N}$, conditional on the value of the parent(s) of X_n , the distribution of X_n is probabilistically independent of the non-parents,

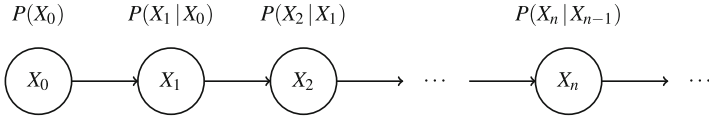


Fig. 5.5 Bayesian network representation of a discrete-time Markov chain $\{X_t\}_{t \in \mathbb{N}_0}$. Nodes represent random variables. An incoming arc on a node represents that the distribution of the corresponding random variable is influenced by the originating node of that arc. Correspondingly, each node associates a probability distribution to its random variable, conditional on the values of the random variables of the nodes on which it is dependent as before

non-descendants of X_n . This is the general interpretation of the independence properties of the arcs in a BN. In the special case of Markov chains that we are considering here, the interpretation vastly simplifies. Notably, the ‘non-parents, non-descendants’ of any node X_n are exactly its ‘grandparents’, ‘great-grandparents’ and so on; it is the set of nodes $\{X_m : m \in \mathbb{N}_0, m < n - 1\}$.

Put differently, the value of X_n influences the distribution of *all* of its descendants (i.e. the nodes $X_m, m > n$), so long as we do not know the value of any of those descendants themselves. We will next consider how we can quantify this.

We start by observing that for each node $X_n, n \in \mathbb{N}$, we have the associated conditional probability $P(X_n | X_{n-1})$. Since the state-space \mathcal{X} is taken to be finite, we can conveniently represent these conditional probabilities in a $|\mathcal{X}| \times |\mathcal{X}|$ matrix. For any $t \in \mathbb{N}_0$, this matrix T_t is defined, for all $x, y \in \mathcal{X}$, as

$$T_t(x, y) = P(X_{t+1} = y | X_t = x), \tag{5.3}$$

where the indexing is taken to be row-first. This matrix T_t is called the *transition matrix* of the Markov chain at time t . Its elements $T_t(x, y)$ are called the *transition probabilities from x to y* , and they are the probabilities that a system that is in state x at time t will be in state y at time $t + 1$. This explains the subscript-indexing, whereby the matrix T_t contains the conditional probabilities associated to node X_{t+1} .

These transition matrices make it easy to connect back to the probability tree representation of Markov chains that we encountered earlier:

Proposition 5.1 *Let $(\mathcal{X}_{\square}^*, <, p_{(\cdot)})$ be a probability tree that is a Markov chain, and let T_t denote the associated family of transition matrices, as defined above. Then for all $t \in \mathbb{N}$ and all $w \in \mathcal{X}^*$ such that $|w| = t$, it holds that $p_w(y) = T_t(w_{\top}, y)$ for all $y \in \mathcal{X}$.*

Proof Use Eq. (5.2), Definition 5.5 and Eq. (5.3). □

The reason that we represent these probabilities using matrices is that this opens up the entire toolbox of linear algebra. We will see that this allows us to very succinctly write down certain relations and properties. For instance, we can now write the influence of a node on its descendants, using a simple matrix product:

Proposition 5.2 *Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time Markov chain, and let T_t be the associated family of transition matrices, as defined above. Then for all $s, t \in \mathbb{N}_0$ such that $s \leq t$, and all $x, y \in \mathcal{X}$, it holds that $P(X_{t+1} = y | X_s = x) = [T_s \cdots T_t](x, y)$.*

Proof We give a proof by induction. For $t = s$ the result is immediate from the definition of the transition matrix T_s . Now suppose the result is true for $t - 1$; we show that it is also true for t :

$$\begin{aligned} P(X_{t+1} = y | X_s = x) &= \sum_{z \in \mathcal{X}} P(X_{t+1} = y, X_t = z | X_s = x) \\ &= \sum_{z \in \mathcal{X}} P(X_t = z | X_s = x) P(X_{t+1} = y | X_t = z, X_s = x) \\ &= \sum_{z \in \mathcal{X}} [T_s \cdots T_{t-1}](x, z) P(X_{t+1} = y | X_t = z) \\ &= \sum_{z \in \mathcal{X}} [T_s \cdots T_{t-1}](x, z) T_t(z, y) = [T_s \cdots T_{t-1} T_t](x, y), \end{aligned}$$

where the first and second equalities are basic properties of probabilities, the third equality is due to the induction hypothesis and the Markov property (c.f. Definition 5.5), the fourth equality uses the definition of the transition matrix T_t and the final equality uses the definition of a matrix product. \square

Another useful property of this representation is that it allows us to write conditional expectations of functions $f \in \mathcal{L}(\mathcal{X})$ using matrix-vector products. In particular, again because \mathcal{X} is finite, any $f \in \mathcal{L}(\mathcal{X})$ can be interpreted as a vector in $\mathbb{R}^{|\mathcal{X}|}$; the coordinates are simply the values $f(x)$, $x \in \mathcal{X}$. Hence:

Proposition 5.3 *Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time Markov chain, and let T_t be the associated family of transition matrices. Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $t \in \mathbb{N}_0$ and all $x \in \mathcal{X}$, it holds that $\mathbb{E}[f(X_{t+1}) | X_t = x] = [T_t f](x)$.*

Proof Simply use the definition of the matrix-vector product:

$$[T_t f](x) = \sum_{y \in \mathcal{X}} T_t(x, y) f(y) = \sum_{y \in \mathcal{X}} P(X_{t+1} = y | X_t = x) f(y) = \mathbb{E}[f(X_{t+1}) | X_t = x].$$

\square

The above properties can be combined to give a simplified version of the law of iterated expectation (Theorem 5.1) that we encountered in Sect. 5.2.1:

Corollary 5.3 *Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time Markov chain, and let T_t be the associated family of transition matrices. Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $s, t \in \mathbb{N}_0$ such that $s \leq t$ and all $x \in \mathcal{X}$, it holds that $\mathbb{E}[f(X_{t+1}) | X_s = x] = [T_s \cdots T_t f](x)$.*

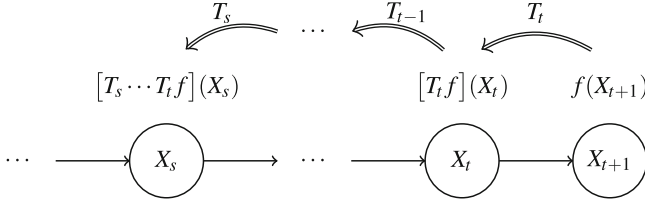


Fig. 5.6 Graphical representation of the ‘pulling back’ interpretation of the simplified version of the law of iterated expectation in Corollary 5.3. The function f , of which we want to compute the expectation on X_{t+1} , given X_s , starts at node X_{t+1} , where its value is trivial. The function is then ‘pulled back’ to the parent X_t of X_{t+1} , by taking the local expectation, by left-multiplying with T_t . This new function $T_t f$ on X_t is then ‘pulled’ back by multiplying with T_{t-1} and so forth. Eventually, the function $T_{s+1} \cdots T_t f$ is pulled into X_s , by left-multiplying with T_s . The resulting function on X_s is the conditional expectation of interest as before

Proof Immediate from Propositions 5.2 and 5.3. □

Note that, where the law of iterated expectation in Theorem 5.1 could be interpreted as ‘pulling back’ in the associated probability tree, the above simplified version can additionally be interpreted as ‘pulling back’ the conditional expectations in the associated BN, through the product of the transition matrices. This is graphically represented in Fig. 5.6.

5.2.3 Transition Graphs

We now move on to yet another graphical representation: the *transition graph* of a homogeneous (discrete-time) Markov chain. We start by noticing the following:

Proposition 5.4 *Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain, and let T_t be the associated family of transition matrices. Then there is a unique matrix T such that $T_t = T$ for all $t \in \mathbb{N}_0$.*

Proof The matrix of interest can be identified as $T \doteq T_0$. Now, using the definition of a homogeneous Markov chain (Definition 5.6) and the transition matrix T_t for any $t \in \mathbb{N}_0$, it holds for all $x, y \in \mathcal{X}$ that

$$\begin{aligned} T(x, y) &= T_0(x, y) = P(X_1=y \mid X_0=x) \\ &= P(X_{(t+1)-t}=y \mid X_0=x) = P(X_{t+1}=y \mid X_t = x) = T_t(x, y), \end{aligned}$$

which concludes the proof; uniqueness is trivial. □

As an aside, note therefore that a discrete-time homogeneous Markov chain can be characterised (up to the initial distribution $P(X_0)$) by a single transition matrix T . In particular, this T can be seen as the canonical parameter of the Markov chain. This

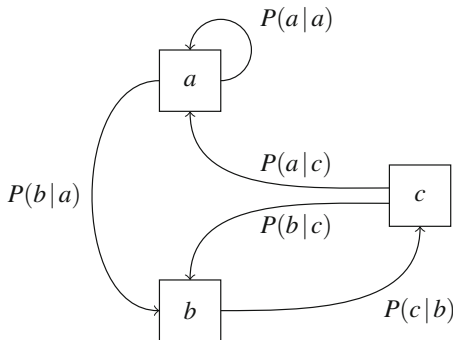


Fig. 5.7 Example transition graph for a discrete-time homogeneous Markov chain with a ternary state-space $\mathcal{X} = \{a, b, c\}$. The transition graph is a directed graph, with a vertex for each state and an arc from the vertex of x to that of y , with $x, y \in \mathcal{X}$, whenever $T(x, y) = P(X_1 = y | X_0 = x) > 0$. The arcs are labelled with the corresponding transition probabilities. The figure uses the shorthand notation $P(y|x)$ for the elements $T(x, y)$ of T as before

relative ease of parameterisation—compared to say an arbitrary stochastic process, which needs separate parameters for every possible history—is arguably one of the reasons that make homogeneous Markov chains such convenient and widely used models.

Moving on, the transition graph of a discrete-time homogeneous Markov chain is a graphical representation of its associated transition matrix T . In this way, this representation emphasises the interactions between the *states*, rather than the random variables. An example transition graph is shown in Fig. 5.7. The formal definition is as follows:

Definition 5.7 (Transition graph) Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain, and let T be its associated transition matrix. Then its associated *transition graph* is a directed graph (V, E) with one vertex for each state, $V = \mathcal{X}$, and, for all $x, y \in \mathcal{X}$, an arc $(x, y) \in E$ whenever $T(x, y) > 0$.

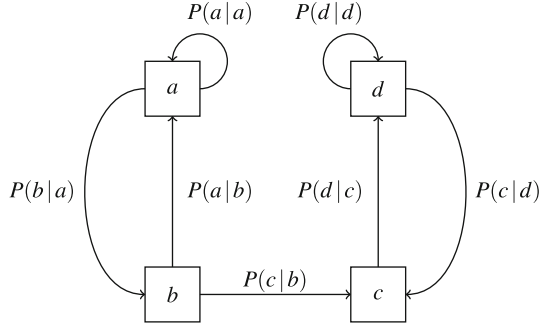
One of the reasons transition graphs are sometimes useful is that they allow one to study which parts of a system can be reached from other parts of the system. The simplest application is that of *communicating states*:

Definition 5.8 (Communicating states) Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain, and let T be its associated transition matrix. For any two states $x, y \in \mathcal{X}$, y is said to be *accessible* from x if there is some $n \in \mathbb{N}$ such that $T^n(x, y) > 0$. Furthermore, x and y are said to *communicate* if y is accessible from x , and x is accessible from y .

Note that in the above, the term T^n denotes the n -th matrix power of T (c.f. Proposition 5.2). This has an intuitive graphical interpretation:

Corollary 5.4 Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain. Then for any $x, y \in \mathcal{X}$, y is accessible from x if and only if there is a path from x to y

Fig. 5.8 Transition graph of a Markov chain that is *not* irreducible. It has two communication classes, $\{a, b\}$ and $\{c, d\}$. The set $\{c, d\}$ dominates $\{a, b\}$ and is the top (communication) class of the Markov chain. This Markov chain is top class regular



in the associated transition graph. Furthermore, x and y communicate if and only if there is a cycle in the associated transition graph that contains both x and y .

Proof Trivial from Definitions 5.7 and 5.8. □

Inspection of the transition graph in Fig. 5.7 shows that, in that example, all states communicate with each other. When this is the case, i.e. when all states communicate, the Markov chain is said to be *irreducible*. A maximal set of states that all communicate with each other is called a *communication class*. Hence, an irreducible Markov chain has only a single communication class, which is equal to \mathcal{X} .

Note that not every Markov chain is irreducible; in general there may be more than one communication class. An example is given in Fig. 5.8. When a communication class $\mathcal{A} \subset \mathcal{X}$ is accessible from a different communication class $\mathcal{B} \subset \mathcal{X}$, then \mathcal{A} is said to *dominate* \mathcal{B} . A communication class which is not dominated is called *maximal*. When a Markov chain has only a single maximal communication class, this is called the *top (communication) class*.

Investigation of the communicating states in a Markov chain is often useful when one is interested in the long-term behaviour of the system. After all, while a system might begin in one state, it need not necessarily always eventually return to that state; this is the property that is illustrated in Fig. 5.8.

An important concept is that of the *regularity* of the communication classes of a Markov chain. A communication class is regular if there is a number $n \in \mathbb{N}$ such that it is possible to go from any state in the class to any other state in the class, in exactly n steps. Of particular importance is the notion of *top class regularity*:

Definition 5.9 (Top class regularity) Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain, and let T be its associated transition matrix. Then the Markov chain is said to be *top class regular* if

$$\{y \in \mathcal{X} : (\exists n \in \mathbb{N})(\forall x \in \mathcal{X}) T^n(x, y) > 0\} \neq \emptyset,$$

and in that case the top class \mathcal{X}_{top} of the Markov chain exists and is equal to this set. When furthermore $\mathcal{X}_{\text{top}} = \mathcal{X}$, the Markov chain itself is said to be *regular*.

The reason that this property is so important is that it provides a sufficient condition for the long-term behaviour of a Markov chain to converge to a stationary distribution, regardless of the state in which it started:

Theorem 5.2 *Let $\{X_t\}_{t \in \mathbb{N}_0}$ be a discrete-time homogeneous Markov chain, and let T be its associated transition matrix. Let this Markov chain be regular. Then there is a probability mass function $P_\infty : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ such that, for all $x, y \in \mathcal{X}$,*

$$P_\infty(y) = \lim_{n \rightarrow +\infty} T^n(x, y).$$

5.3 Imprecise Discrete-Time Markov Chains

We will now move on to the discussion surrounding *imprecise (discrete-time) Markov chains* (IDTMCs). So, we still consider the time-dimension $\mathbb{T} = \mathbb{N}_0$. We will generalise each of the representations that we previously encountered to this new setting, where we roughly follow the same order as in Sect. 5.2.

So, let us start with the ‘measure-theoretic’ representation of imprecise stochastic processes. In this setting, we consider a set \mathbb{P} of probability measures on the measurable space of paths (Ω, \mathcal{F}) . Then for each $P \in \mathbb{P}$, we have a probability space (Ω, \mathcal{F}, P) , to which we can associate the precise stochastic process $\{X_t\}_{t \in \mathbb{N}_0}$ as in Definition 5.1. For any function $f \in \mathcal{L}(\mathcal{X}^n)$, $n \in \mathbb{N}$, we can express the expected value on the n time points $\mathbf{t} \subset \mathbb{N}_0$ as $\mathbb{E}_P[f(X_{\mathbf{t}})]$ as in Sect. 5.2. Recall from Chap. 2 that in this imprecise probabilistic context, we are more generally interested in the *lower* and *upper expectation* of f , which are defined, respectively, as

$$\underline{\mathbb{E}}_{\mathbb{P}}[f(X_{\mathbf{t}})] \doteq \inf_{P \in \mathbb{P}} \mathbb{E}_P[f(X_{\mathbf{t}})] \quad \text{and} \quad \overline{\mathbb{E}}_{\mathbb{P}}[f(X_{\mathbf{t}})] \doteq \sup_{P \in \mathbb{P}} \mathbb{E}_P[f(X_{\mathbf{t}})].$$

We briefly recall the well-known conjugacy relation $\overline{\mathbb{E}}_{\mathbb{P}}[f(X_{\mathbf{t}})] = -\underline{\mathbb{E}}_{\mathbb{P}}[-f(X_{\mathbf{t}})]$, from which it follows that we can present the remainder of this discussion entirely in terms of lower expectations; any corresponding results on upper expectations follow directly through this relation.

Slightly more generally than the above, we will focus on *conditional* lower expectations. Similar to the precise case that we discussed before, these are defined for any $f \in \mathcal{L}(\mathcal{X}^{n+m})$, $n, m \in \mathbb{N}$, any $\mathbf{s}, \mathbf{t} \subset \mathbb{N}_0$ such that \mathbf{s} and \mathbf{t} are of length n and m , respectively, and any $x_{\mathbf{s}} \in \mathcal{X}^n$, as

$$\underline{\mathbb{E}}_{\mathbb{P}}[f(X_{\mathbf{s}}, X_{\mathbf{t}}) \mid X_{\mathbf{s}} = x_{\mathbf{s}}] \doteq \inf_{P \in \mathbb{P}} \mathbb{E}_P[f(X_{\mathbf{s}}, X_{\mathbf{t}}) \mid X_{\mathbf{s}} = x_{\mathbf{s}}],$$

whenever $\underline{\mathbb{E}}_{\mathbb{P}}[\mathbb{I}_{x_{\mathbf{s}}}(X_{\mathbf{s}})] > 0$. In this last condition, $\mathbb{I}_{x_{\mathbf{s}}}$ is the indicator of $x_{\mathbf{s}}$; for all $y_{\mathbf{s}} \in \mathcal{X}^n$, $\mathbb{I}_{x_{\mathbf{s}}}(y_{\mathbf{s}}) \doteq 1$ if $x_{\mathbf{s}} = y_{\mathbf{s}}$ and $\mathbb{I}_{x_{\mathbf{s}}}(y_{\mathbf{s}}) \doteq 0$, otherwise. Note that then

$$0 < \underline{\mathbb{E}}_{\mathbb{P}}[\mathbb{I}_{x_s}(X_s)] = \inf_{P \in \mathbb{P}} \mathbb{E}_P[\mathbb{I}_{x_s}(X_s)] = \inf_{P \in \mathbb{P}} P(X_s = x_s),$$

so this condition guarantees that the conditional expectations are well-defined for all the precise measures $P \in \mathbb{P}$. As before, there are formalisms where this condition is not strictly required—see, for example, the discussion around the local models of probability trees—or where it can be weakened. For simplicity, we keep the condition here to ensure that everything remains well-defined also under the measure-theoretic interpretation.

We are now ready to give the formal definition of an imprecise discrete-time Markov chain (IDTMC):

Definition 5.10 (IDTMC as set of processes) An *imprecise discrete-time Markov chain* is a set \mathbb{P} of probability measures on the measurable space (Ω, \mathcal{F}) , with associated lower expectation operator $\underline{\mathbb{E}}_{\mathbb{P}}$ as defined above, such that, for all $f \in \mathcal{L}(\mathcal{X})$ and all $s_1, \dots, s_n, t \in \mathbb{N}_0$ such that $s_1 < \dots < s_n < t$,

$$\underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_{s_1}, \dots, X_{s_n}] = \underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_{s_n}].$$

Furthermore, an imprecise discrete-time Markov chain is called *homogeneous* if, for all $s, t \in \mathbb{N}_0$, $s < t$, and all $f \in \mathcal{L}(\mathcal{X})$, it holds that $\underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_s] = \underline{\mathbb{E}}_{\mathbb{P}}[f(X_{t-s}) \mid X_0]$.

Let us compare this with Definition 5.5, the measure-theoretic definition of a precise Markov chain. The first difference is that the imprecise definition above is phrased in terms of (lower) expectations, whereas the precise definition used probabilities. We recall that this is because, in the framework of imprecise probability, it does not suffice to state results in terms of (lower) probabilities; instead the more general language of (lower) expectation operators is required.

Nevertheless, this definition implies that, in terms of lower probabilities,

$$\begin{aligned} \inf_{P \in \mathbb{P}} P(X_t = x \mid X_{s_1}, \dots, X_{s_n}) &= \underline{\mathbb{E}}_{\mathbb{P}}[\mathbb{I}_x(X_t) \mid X_{s_1}, \dots, X_{s_n}] \\ &= \underline{\mathbb{E}}_{\mathbb{P}}[\mathbb{I}_x(X_t) \mid X_{s_n}] = \inf_{P \in \mathbb{P}} P(X_t = x \mid X_{s_n}), \end{aligned}$$

which displays this imprecise Markov condition in more familiar terms.

One may wonder at this point whether an imprecise Markov chain \mathbb{P} is itself a set of Markov chains; the answer to this question is a resounding *no* (or at least, not necessarily). This point deserves the strongest possible emphasis:

An element of an imprecise Markov chain \mathbb{P} need **not** be a Markov chain! So, in general $P(X_t \mid X_{s_1}, \dots, X_{s_n}) \neq P(X_t \mid X_{s_n})$ for $P \in \mathbb{P}$, with $s_1 < \dots < s_n < t$ in \mathbb{N}_0 .

To clarify, the ‘imprecise Markov condition’ of an imprecise Markov chain is an ‘independence’ assessment about the *lower envelope* only. Formally, it is an assessment of *epistemic irrelevance*—a specific type of independence that arises in

imprecise probability theory—which is weaker than *strong independence* a different type of independence, and what would hold of all $P \in \mathbb{P}$ were Markov chains.

In a similar vein, the notion of homogeneity is here only enforced on the lower envelope. So, for an IDTMC \mathbb{P} that is homogeneous, there may be processes $P \in \mathbb{P}$ that are neither Markov nor homogeneous.

The reason why we stress this so strongly is twofold. First of all, it implies that the structural assumptions of an imprecise Markov chain are in fact much weaker than those of a precise Markov chain—we no longer assume that future events are fully independent of the history, given the current state, or that their distribution is independent of the point in time. They might be, of course—there *are* elements $P \in \mathbb{P}$ that satisfy those properties—but it's not enforced as strictly. In other words, this model also represents 'higher-order' uncertainty about the *structural properties* of the system that we are trying to model.

The second reason is that this property is central to all the efficient computational methods that have been developed for working with imprecise Markov chains. We will next illustrate this point by moving the discussion to the representation of IDTMCs as *imprecise probability trees*.

5.3.1 Imprecise Probability Trees

Recall that for precise probability trees, we associate with each situation $w \in \mathcal{X}_{\square}^*$ a local model p_w , which is a probability mass function on \mathcal{X} . In contrast, in order to define *imprecise* probability trees, we will consider *imprecise local models*. Such an imprecise local model \mathcal{P}_w is simply a *set* of probability mass functions on \mathcal{X} . This leads to the following definition:

Definition 5.11 (Imprecise probability tree) An imprecise probability tree is a tuple $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$, where $(\mathcal{X}_{\square}^*, \prec)$ is an event tree and $\mathcal{P}_{(\cdot)}$ is a set-valued function such that, for all $w \in \mathcal{X}_{\square}^*$, \mathcal{P}_w is a non-empty set of probability mass functions on \mathcal{X} .

An obvious question is how one should interpret such imprecise probability trees. As a first step, we consider the (precise) probability trees that are *compatible* with a given imprecise probability tree:

Definition 5.12 Let $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$ be an imprecise probability tree. Then a (precise) probability tree $(\mathcal{X}_{\square}^*, \prec, p_{(\cdot)})$ is called *compatible* with this imprecise probability tree, if $p_w \in \mathcal{P}_w$ for all $w \in \mathcal{X}_{\square}^*$.

This immediately lets us connect back to the sets-of-measures that we discussed before. Specifically, consider an imprecise probability tree $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$, and suppose the tree $(\mathcal{X}_{\square}^*, \prec, p_{(\cdot)})$ is compatible with it. Then, using the method outlined in Sect. 5.2.1, we can associate a (precise) measure P to this precise tree. Collecting in the set \mathbb{P} all the associated measures of all precise trees that are compatible with the imprecise tree, we obtain a set representation as in Sect. 5.3.

The connection in the other direction is analogous but a bit more subtle. In particular, if we start from an IDTMC \mathbb{P} , then each $P \in \mathbb{P}$ induces a precise probability tree. Using the local models of this tree, we can construct set-valued local models by simply varying P over \mathbb{P} . These set-valued local models can then be used to construct an imprecise probability tree. Clearly, there are then precise trees that are compatible with this imprecise tree, and each such precise tree induces a precise measure P' . However, and this is the crucial observation, it is in general *not* guaranteed that such P' are included in \mathbb{P} !

As a simple example, suppose that $\mathcal{X} = \{a, b\}$ and we start with a set \mathbb{P} containing only two i.i.d. processes, whose local models are given by p, h , respectively. Then, the induced imprecise probability tree has local models $\mathcal{P}_w = \{p, h\}$ for all $w \in \mathcal{X}_{\square}^*$. On the other hand, we can easily construct a non-i.i.d. process such that, for all $w \in \mathcal{X}_{\square}^*$, its local model is $p_w = p$ if $w_{\top} = a$ and $p_w = h$, otherwise. Then clearly this process was not in the original set \mathbb{P} , but it is compatible with the imprecise probability tree.

To prevent this from happening, we will require that the set representation \mathbb{P} of the IDTMC is ‘large enough’. Specifically, what we need is that it is already closed under such ‘recombination’ of local models at different points in time. Whenever this property holds, we will say that the IDTMC is *separately specified*. Clearly, when we start from an imprecise probability tree and construct its set of compatible processes, this IDTMC will then satisfy this property. In the remainder of this section, we will assume that a given set \mathbb{P} is indeed separately specified. Further on, when we consider the parametrisation of an IDTMC, we will consider an easy condition that ensures this will hold.

With this connection between the two representations in place, we can again start to consider computational methods for lower expectations. Analogous to what we have seen before, in this context we have a *law of iterated lower expectation* that we can use as a computational tool. The imprecise probability tree representation again provides graphical intuition.

Similar to the exposition in Sect. 5.2.1, we start with a function $f \in \mathcal{L}(\mathcal{X}^{n+1})$ of which we want to compute the lower expectation with respect to the states at the time points $0, \dots, n$. Then for any situation $w \in \mathcal{X}^*$ such that $|w| = n + 1$, the lower expectation is trivial:

$$\mathbb{E}_{\mathbb{P}} [f(X_{0:n}) \mid X_{0:n} = w] = f(w).$$

We then again ‘pull back’ to the parent situation v of w ; this is where the main difference with Sect. 5.2.1 occurs. Notably, we here have an *imprecise* local model \mathcal{P}_v associated to this node v . The point to the law of iterated lower expectation is that it suffices to only compute the associated conditional lower expectation locally:

$$\mathbb{E}_{\mathbb{P}} [f(X_{0:n}) \mid X_{0:(n-1)}=v] = \inf_{p_v \in \mathcal{P}_v} \sum_{x \in \mathcal{X}} p_v(x) \mathbb{E}_{\mathbb{P}} [f(X_{0:n}) \mid X_{0:(n-1)}=v, X_n=x].$$

Exactly analogous to the precise case, by repeatedly pulling back until we reach the root of the tree, we eventually compute

$$\mathbb{E}_{\mathbb{P}} [f(X_{0:n})] = \inf_{P_{\square} \in \mathcal{P}_{\square}} \sum_{x \in \mathcal{X}} p_{\square}(x) \mathbb{E}_{\mathbb{P}} [f(X_{0:n}) \mid X_0 = x],$$

which is the lower expectation of interest.

As before, the need to specify these (imprecise) local models \mathcal{P}_w for all situations $w \in \mathcal{X}_{\square}^*$ makes such a model difficult to work with. This is simplified for imprecise Markov chains; note that we here assume the analogue of homogeneity to hold implicitly:

Definition 5.13 (Homogeneous IDTMC as imprecise probability tree) An imprecise probability tree $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$ is called an imprecise homogeneous discrete-time Markov chain if $\mathcal{P}_v = \mathcal{P}_w$ for all $v, w \in \mathcal{X}^*$ for which $v_{\top} = w_{\top}$.

Corollary 5.5 *Let $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$ be a homogeneous IDTMC. Then $\mathcal{P}_w = \mathcal{P}_x$ for all $x \in \mathcal{X}$ and all $w \in \mathcal{X}^*$ such that $w_{\top} = x$.*

Proof Trivial from Definition 5.13 and the fact that all $x \in \mathcal{X}$ are also situations. □

As above, an IDTMC $(\mathcal{X}_{\square}^*, \prec, \mathcal{P}_{(\cdot)})$ has a set of compatible precise probability trees, each of which induces a measure P , and these are collected in the set \mathbb{P} , which is the measure-theoretic IDTMC representation from Definition 5.10. Observe that a precise probability tree does *not* have to be a (homogeneous) Markov chain, for it to be compatible with a given IDTMC! That is, to be compatible, each local model p_w , $w \in \mathcal{X}_{\square}^*$, should be in the set $\mathcal{P}_{w_{\top}}$, and this set depends only on the most recent state w_{\top} of the situation w . But, while in a different situation v such that $v_{\top} = w_{\top}$, we do require that $p_v \in \mathcal{P}_{v_{\top}} = \mathcal{P}_{w_{\top}}$; we do *not* require that $p_v = p_w$!

We will next illustrate that the law of iterated lower expectation simplifies further for imprecise Markov chains. We do this again by considering the imprecise counterpart of Bayesian networks.

5.3.2 Credal Networks

We here consider the graphical representation of imprecise Markov chains as *credal networks*. This is the imprecise generalisation of the Bayesian network representation that we encountered in Sect. 5.2.2. The graphical structure is as before, with the notable differences being (i) the local models (which are here replaced with imprecise local models) and (ii) the interpretation of the independence properties induced by the arcs. Regarding the second point, it suffices for our present purpose to note that we interpret the structure as a credal network under epistemic irrelevance. This then has the same consequence as that stated in the beginning of Sect. 5.3: given the value of the parent of a node X_t , $t \in \mathbb{N}_0$, the lower expectation

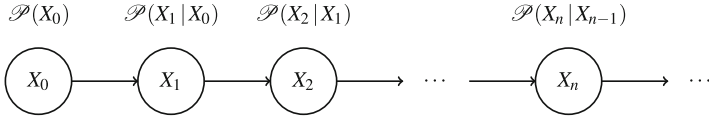


Fig. 5.9 Credal network representation of an imprecise discrete-time Markov chain. An incoming arc on a node represents that the local uncertainty model of the corresponding variable is influenced by the originating node of that arc. Correspondingly, each node associates an imprecise probability model to its variable, conditional on the values of the random variables of the nodes on which it is dependent

of any function dependent on X_t does not depend on the values of the non-parents, non-descendants (again, grandparents and so on) of X_t . For reference, the graphical representation is drawn in Fig. 5.9.

The interpretation in terms of sets of distributions is as would be expected; the model induces a set \mathbb{P} , each $P \in \mathbb{P}$ of which satisfies $P(X_n | X_{n-1}) \in \mathcal{P}(X_n | X_{n-1})$ for all $n \in \mathbb{N}$, and $P(X_0) \in \mathcal{P}(X_0)$. As before, the independence assumptions are not necessarily required to hold for these compatible precise models. Conversely, if we are given an IDTMC \mathbb{P} , then the local models $\mathcal{P}(X_n | X_{n-1})$ of the credal network are constructed by restricting attention to the conditional events $P(X_n | X_{n-1})$ and varying P over \mathbb{P} .

Similar to the discussion around the interpretation of imprecise probability trees, we here also need some ‘closedness’ assumptions to ensure this duality of representations holds. Specifically, we again require that \mathbb{P} is separately specified. Furthermore, it is assumed that the local models $\mathcal{P}(X_n | X_{n-1})$ of the credal network have *separately specified rows*. This means that these local models are not arbitrary sets of conditional probabilities. If we let $\mathcal{P}(X_n | X_{n-1} = x) \doteq \{P(X_n | X_{n-1} = x) \in \mathcal{P}(X_n | X_{n-1})\}$ for all $x \in \mathcal{X}$, then what we require is that

$$\mathcal{P}(X_n | X_{n-1}) = \times_{x \in \mathcal{X}} \mathcal{P}(X_n | X_{n-1} = x). \tag{5.4}$$

Under these conditions, we can straightforwardly switch between representations.

We next generalise the exposition in Sect. 5.2.2 regarding the associated transition matrices. To this end, fix any $t \in \mathbb{N}_0$. Then, as in the precise case, each element $P(X_{t+1} | X_t) \in \mathcal{P}(X_{t+1} | X_t)$ induces a transition matrix T_t . So, let us now consider the set \mathcal{T}_t of transition matrices that is induced by the imprecise local models:

$$\mathcal{T}_t \doteq \left\{ T_t : (\forall x, y \in \mathcal{X} : T_t(x, y) = P(X_{t+1} = y | X_t = x)), \right. \\ \left. P(X_{t+1} | X_t) \in \mathcal{P}(X_{t+1} | X_t) \right\}.$$

A key insight is that we can use this set of transition matrices to define a convenient computational tool for lower expectations:

Definition 5.14 Let \mathbb{P} be an IDTMC, and let \mathcal{T}_t be the associated family of sets of transition matrices, as defined above. Then, for each $t \in \mathbb{N}_0$, the associated *lower transition operator* $\underline{T}_t : \mathcal{L}(\mathcal{X}) \rightarrow \mathcal{L}(\mathcal{X})$ is defined, for all $f \in \mathcal{L}(\mathcal{X})$ and all $x \in \mathcal{X}$, as

$$[\underline{T}_t f](x) \doteq \inf_{T_t \in \mathcal{T}_t} [T_t f](x).$$

This lower transition operator essentially fulfils the same role as the transition matrices from which it is derived. In particular, we have the following:

Proposition 5.5 *Let \mathbb{P} be an IDTMC, and let \underline{T}_t be the associated family of lower transition operators. Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $t \in \mathbb{N}_0$ and all $x \in \mathcal{X}$, it holds that*

$$[\underline{T}_t f](x) = \mathbb{E}_{\mathbb{P}}[f(X_{t+1}) \mid X_t = x].$$

Proof Simply use the definitions together with Proposition 5.3:

$$\begin{aligned} [\underline{T}_t f](x) &= \inf_{T_t \in \mathcal{T}_t} [T_t f](x) = \inf_{T_t \in \mathcal{T}_t} \sum_{y \in \mathcal{X}} f(y) T_t(x, y) \\ &= \inf_{P(X_{t+1} \mid X_t) \in \mathcal{P}(X_{t+1} \mid X_t)} \sum_{y \in \mathcal{X}} f(y) P(X_{t+1}=y \mid X_t = x) \\ &= \inf_{P \in \mathbb{P}} \sum_{y \in \mathcal{X}} f(y) P(X_{t+1}=y \mid X_t = x) \\ &= \inf_{P \in \mathbb{P}} \mathbb{E}_P[f(X_{t+1}) \mid X_t = x] = \mathbb{E}_{\mathbb{P}}[f(X_{t+1}) \mid X_t = x], \end{aligned}$$

where in the fourth equality, we used the definition of the compatible measures. \square

As in Corollary 5.3, we can now state the simplified law of iterated lower expectation for imprecise Markov chains, using these lower transition operators:

Theorem 5.3 *Let \mathbb{P} be an IDTMC that is separately specified, and let \underline{T}_t be the associated family of lower transition operators. Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $s, t \in \mathbb{N}_0$ such that $s \leq t$ and all $x \in \mathcal{X}$, it holds that*

$$\mathbb{E}_{\mathbb{P}}[f(X_t) \mid X_s = x] = [\underline{T}_s \cdots \underline{T}_t f](x),$$

where the right-hand side represents an iterated operator product (composition).

We omit the full proof, but the interested reader can reconstruct the argument by using the general computational process of iterated lower expectation as explained in Sect. 5.3.1, the imprecise Markov property from Definition 5.10 and the interpretation of the lower transition operator from Proposition 5.5.

5.3.3 Limits of Homogeneous IDTMCs

We conclude the discussion of imprecise discrete-time Markov chains with some results about their limit behaviour, in analogy to the results in Sect. 5.2.3. We start again by restricting attention to homogeneous IDTMCs, and notice the following (we omit the proof, which is straightforward):

Proposition 5.6 *Let \mathbb{P} be a homogeneous IDTMC, and let \underline{T}_t be the associated family of lower transition operators. Then there is a unique lower transition operator $\underline{T} : \mathcal{L}(\mathcal{X}) \rightarrow \mathcal{L}(\mathcal{X})$, such that, for all $f \in \mathcal{L}(\mathcal{X})$, $\underline{T}_t f = \underline{T} f$ for all $t \in \mathbb{N}_0$.*

We take a moment here to remark on a property that was already encountered in Chap. 2: the duality between lower expectation operators and closed and convex sets of probability measures. Indeed, this correspondence was also used in Definition 5.14 above, where we used the sets \mathcal{T}_t of transition matrices, to construct the lower transition operator \underline{T}_t . Since, as we have just seen, the dynamics of a homogeneous IDTMC can be completely described by a single \underline{T} , it now makes sense to think about the other direction.

Specifically, corresponding to \underline{T} , there exists a closed and convex set \mathcal{T} of transition matrices, such that $\underline{T}f = \inf_{T \in \mathcal{T}} Tf$ for all $f \in \mathcal{L}(\mathcal{X})$. This implies that (up to the initial distribution at time zero) an IDTMC can also be characterised by such a set \mathcal{T} . So, whereas we noted in Sect. 5.2.2 that a (precise) discrete-time Markov chain's canonical parameter is a single transition matrix T , for a homogeneous IDTMC, the parameter can be understood as a single closed and convex set \mathcal{T} of transition matrices. Moreover, if in this parametrisation we ensure that \mathcal{T} has *separately specified rows*—essentially, satisfies a property exactly analogous to Eq. (5.4)—then the corresponding IDTMC will also be separately specified.

Furthermore, in Sect. 5.2.2 we used a property of the associated transition matrix T , to state a sufficient condition for the long-term behaviour of the Markov chain to converge to a distribution over the states, independently of the state in which it started. We here have a similar result, which starts by introducing the conjugate upper transition operator $\bar{T} : \mathcal{L}(\mathcal{X}) \rightarrow \mathcal{L}(\mathcal{X}) : f \mapsto -\underline{T}(-f)$.

Now, recall that in the precise case, a homogeneous discrete-time Markov chain with transition matrix T was said to be *regular*, if there was some $n \in \mathbb{N}$ such that $T^n(x, y) > 0$ for all $x, y \in \mathcal{X}$. The interpretation is clear: the Markov chain is regular if and only if there is some finite number of steps n in which every state x can reach every state y . This is now generalised to the imprecise case:

Definition 5.15 (Regularity for homogeneous IDTMC) Let \mathbb{P} be a homogeneous IDTMC with associated lower (and upper) transition operator \underline{T} (and \bar{T}). Then the IDTMC is *regular* if there is some $n \in \mathbb{N}$ such that $[\bar{T}^n \mathbb{I}_y](x) > 0$ for all $x, y \in \mathcal{X}$.

Let us consider this definition. One difference with the precise case is the introduction of the indicator function \mathbb{I}_y on the state $y \in \mathcal{X}$; this was introduced because,

in contrast to matrices, we cannot index the ‘elements’ of the transition operator. Specifically, using Theorem 5.3, we can interpret the condition as

$$0 < \left[\overline{T}^n \mathbb{I}_y \right](x) = \overline{\mathbb{E}}_{\mathbb{P}}[\mathbb{I}_y(X_n) \mid X_0 = x] = \sup_{P \in \mathbb{P}} P(X_n = y \mid X_0 = x),$$

for all $x, y \in \mathcal{X}$ and some $n \in \mathbb{N}$. What regularity asks for, then, is for there to be some $n \in \mathbb{N}$ such that is possible for all $x, y \in \mathcal{X}$ to move from x to y in exactly n steps, according to some $P \in \mathbb{P}$. In particular, the (precise) measure P for which this needs to be possible can be different for every pair $x, y \in \mathcal{X}$. Regularity for IDTMCs then is in a sense a much weaker—easier to satisfy—condition than that for precise Markov chains. Nevertheless, the condition is sufficient for the following:

Theorem 5.4 *Let \mathbb{P} be a homogeneous IDTMC that is separately specified and regular, with associated lower transition operator \underline{T} . Then, there is a unique lower expectation operator $\underline{\mathbb{E}}_{\mathbb{P}}[\cdot(X_{+\infty})] : \mathcal{L}(\mathcal{X}) \rightarrow \mathbb{R}$ such that, for all $f \in \mathcal{L}(\mathcal{X})$ and all $x \in \mathcal{X}$,*

$$\underline{\mathbb{E}}_{\mathbb{P}}[f(X_{+\infty})] = \lim_{n \rightarrow +\infty} \underline{\mathbb{E}}_{\mathbb{P}}[f(X_n) \mid X_0 = x] = \lim_{n \rightarrow +\infty} [\underline{T}^n f](x).$$

Furthermore, this is the unique \underline{T} -invariant lower expectation on $\mathcal{L}(\mathcal{X})$, meaning that $\underline{\mathbb{E}}_{\mathbb{P}}[f(X_{+\infty})] = \underline{\mathbb{E}}_{\mathbb{P}}[\underline{T} f](X_{+\infty})$ for all $f \in \mathcal{L}(\mathcal{X})$.

5.4 Imprecise Continuous-Time Markov Chains

We now move on to the discussion about (imprecise) *continuous-time* Markov chains. We have already encountered this setting several times in the preceding discussions but have generally skipped over any details. Let us recall from Sect. 5.2 that continuous-time stochastic processes are identified with a time-dimension $\mathbb{T} = \mathbb{R}_{\geq 0}$ and that the elements ω of the outcome space of paths Ω are maps $\omega : \mathbb{R}_{\geq 0} \rightarrow \mathcal{X}$. The measure-theoretic definition is then as before, where we consider the abstract probability space (Ω, \mathcal{F}, P) , and the stochastic process $\{X_t\}_{t \in \mathbb{R}_{\geq 0}}$ is a family of random variables on this space. Furthermore, measure-theoretic definitions of (homogeneous) continuous-time Markov chains (CTMCs) have already been encountered in Definitions 5.5 and 5.6.

How, then, can these models be interpreted? Let us start by considering the simplest case, *viz.*, a precise and homogeneous Markov chain in continuous-time. According to the previous definitions, this is a stochastic process such that

1. $P(X_t \mid X_{s_1}, \dots, X_{s_n}) = P(X_t \mid X_{s_n})$ for all $s_1 < \dots < s_n < t$ in $\mathbb{R}_{\geq 0}$, and
2. $P(X_t \mid X_s) = P(X_{t-s} \mid X_0)$ for all $s < t$ in $\mathbb{R}_{\geq 0}$.

The immediate difficulty of moving on from this abstract representation is that the time-dimension is now, in a sense, too big to use any of the previous representations.

For instance, we could try to draw a ‘continuous-time’ probability tree, where the local model of a situation with terminal state w_\top is given by a probability mass function $P(X_t | X_0 = w_\top)$. But what is the time t that we should use? When we were working in discrete-time, the approach was to use the *next* time point, as viewed from the current situation. But of course, there is no ‘next’ time t when working in continuous-time! This difficulty of using graphical representations is the main reason that we have postponed the treatment of continuous-time processes until now, thereby hopefully allowing the reader to first develop some graphical intuition for the discrete-time case.

Nevertheless, all is not lost; the first interpretation that we will consider is to view continuous-time processes as limits of discrete-time ones. To this end, it will be convenient to consider the transition-matrix T associated with a homogeneous DTMC. Let us recall from Sects. 5.2.2 and 5.2.3 that the elements of such a matrix represent the ‘transition probabilities’ of the system, that is, the probability of moving from a state x to a state y , in one time step:

$$T(x, y) = P(X_1 = y | X_0 = x).$$

We can use this formalism to interpret the continuous-time case, by simply ‘fixing the length of the step’. That is, consider some ‘step size’ $\Delta > 0$. Then, for a homogeneous CTMC, we know that

$$P(X_{t+\Delta} | X_t) = P(X_\Delta | X_0),$$

for all $t \in \mathbb{R}_{\geq 0}$, so we can collect these ‘transition probabilities’ in a matrix T_Δ :

$$T_\Delta(x, y) = P(X_\Delta = y | X_0 = x) \quad \text{for all } x, y \in \mathcal{X}.$$

Clearly, the elements of T_Δ are the probabilities for the system to end up in a state y , if it is currently in a state x , after a time duration of Δ has elapsed. Provided, then, that we are not interested in a granularity of the time-dimension that is finer than Δ , this representation suffices. The matrix T_Δ can be associated with a DTMC, and all the previous results can be used. For instance, for any multiple $n \in \mathbb{N}$ of Δ , we use Proposition 5.2 to find that

$$P(X_{n\Delta} = y | X_0 = x) = T_\Delta^n(x, y).$$

But, of course, the point of using the continuous-time representation is that we *are* interested in an arbitrarily fine granularity of the time-dimension. In particular, the measure-theoretic definition encodes this arbitrary granularity, and it seems a waste to only focus on the restriction to a single step size Δ . The ‘trick’, then, is to take the limit as Δ goes to zero, and somehow usefully represent this limit. It is hopefully clear from the above discussion that, as we decrease Δ further and further, the associated transition matrix T_Δ covers increasingly smaller steps along the time-

dimension. And, for each such positive Δ , we can associate a discrete-time Markov chain and use all the previous interpretations that we developed.

We first remark that the naive limit does not encode a lot of information; ignoring possible issues of continuity, it trivially holds that

$$\lim_{\Delta \rightarrow 0^+} P(X_\Delta = y | X_0 = x) = P(X_0 = y | X_0 = x) = \begin{cases} 1 & \text{if } y = x, \text{ and} \\ 0 & \text{otherwise.} \end{cases} \quad (5.5)$$

In matrix notation this reads as $\lim_{\Delta \rightarrow 0^+} T_\Delta = I$, where I denotes the $|\mathcal{X}| \times |\mathcal{X}|$ identity matrix. Colloquially, we might understand this as saying that ‘if time does not evolve, the system does not change’. This is clearly an almost tautological statement to make of what may be interpreted as a dynamical system. So let us consider how the system *does* change as time evolves. The natural representation for this is obviously the *derivative* of the transition matrix T_Δ ; this is the limit interpretation that we shall use. Ignoring technical issues of differentiability, we have

$$\left. \frac{dT_\Delta}{d\Delta} \right|_{\Delta=0} = \lim_{\Delta \rightarrow 0^+} \frac{T_\Delta - I}{\Delta} =: Q, \quad (5.6)$$

where we have used the previous observation that $T_0 = I$. On the right-hand side, the term Q is called the *transition rate matrix* of the homogeneous CTMC (or sometimes simply the *rate matrix*). It is clear from the above definition that it encodes the *rate of change* of the transition probabilities around time zero. It satisfies the following properties:

Definition 5.16 (Transition Rate Matrix) A real-valued $|\mathcal{X}| \times |\mathcal{X}|$ matrix Q is called a *transition rate matrix* if, for all $x \in \mathcal{X}$, it holds that

1. $Q(x, y) \geq 0$ for all $y \in \mathcal{X}$ such that $x \neq y$ and
2. $\sum_{y \in \mathcal{X}} Q(x, y) = 0$.

The elements $Q(x, y)$ of a rate matrix can be interpreted as the ‘speed’ with which the process moves from the state x to the state y . In the above definition, the two conditions imply that the diagonal elements $Q(x, x)$ are always non-positive. On the other hand, the first condition states that the off-diagonal elements are non-negative. Combined this can be understood as saying that the system will move ‘out’ of the current state (the non-positivity of the diagonal elements) and ‘into’ some other states (the non-negativity of the off-diagonals).

A more concrete way to interpret the rate-matrix is through a linearised approximation of the transition probabilities over a small enough time step. That is, it follows from Eq. (5.6) that, for ‘small enough’ $\Delta > 0$, it holds that $Q \approx (T_\Delta - I)/\Delta$; hence also

$$T_\Delta \approx I + \Delta Q. \quad (5.7)$$

We therefore see that the matrix Q can be used to approximately compute the transition probabilities over a small enough time step.

An obvious next question is if we can extrapolate this to compute the matrix T_t that contains the transition probabilities over an arbitrary duration t . Indeed we can, although it requires a bit of setup. For any $t \in \mathbb{R}_{\geq 0}$, first define the transition matrix of the CTMC after time t :

$$T_t(x, y) := P(X_t = y \mid X_0 = x) \quad \text{for all } x, y \in \mathcal{X}.$$

Then we differentiate in t ; to this end, first fix $\Delta > 0$, and use the Markov property and homogeneity to derive that $T_{t+\Delta} = T_t T_\Delta = T_\Delta T_t$ (c.f. Proposition 5.2). Then we proceed by using Eq. (5.6):

$$\frac{dT_t}{dt} = \lim_{\Delta \rightarrow 0^+} \frac{T_{t+\Delta} - T_t}{\Delta} = \lim_{\Delta \rightarrow 0^+} \frac{T_\Delta T_t - T_t}{\Delta} = \left(\lim_{\Delta \rightarrow 0^+} \frac{T_\Delta - I}{\Delta} \right) T_t = QT_t.$$

Using also Eq. (5.5), we can now write the matrix differential equation

$$\frac{dT_t}{dt} = QT_t, \quad T_0 = I,$$

whose solution is the *matrix exponential* of Q t :

$$T_t = e^{Qt}.$$

We recall from Proposition 5.4 that the dynamic behaviour of a homogeneous discrete-time Markov chain can be characterised by a single transition matrix T and that therefore this matrix constitutes the canonical parameter of the process. Because the matrix Q can be used to (re-)construct the transition matrices of a homogeneous CTMC over any time duration, it plays the same role here.

Proposition 5.7 *Let $\{X_t\}_{t \in \mathbb{R}_{\geq 0}}$ be a continuous-time homogeneous Markov chain, with transition rate matrix Q as defined above. Then for all $t \in \mathbb{R}_{\geq 0}$, the transition probabilities $P(X_t = y \mid X_0 = x)$, $x, y \in \mathcal{X}$ after time t are given by the elements $T_t(x, y)$ of the transition matrix $T_t = e^{Qt}$.*

While we do not aim to give a complete treatment on the interpretation of the matrix exponential, some properties are worth pointing out. First of all, it can be defined analogously to the exponential function of real numbers, that is, through a Taylor expansion around zero. Specifically, it holds that

$$T_t = e^{Qt} := \sum_{k=0}^{+\infty} \frac{t^k Q^k}{k!}.$$

Thus, the approximation in Eq. (5.7) can be seen as a first-order truncation of the series above.

As a second important point, we can consider the entire family of transition matrices T_t for all $t \in \mathbb{R}_{\geq 0}$. Then this family constitutes a *semi-group* of transition matrices, and Q is the *generator* of this semi-group. Specifically, it holds that $T_{t+s} = T_t T_s$ for all $t, s \in \mathbb{R}_{\geq 0}$ —this is called the *semi-group* property. Observe that it is analogous to the result in Proposition 5.2 and that we already used this property for the matrix $T_{t+\Delta}$ when constructing the derivative.

These properties immediately yield a different representation for the matrix exponential, which will be convenient further on. We omit the proof.

Proposition 5.8 *Let $\{X_t\}_{t \in \mathbb{R}_{\geq 0}}$ be a continuous-time homogeneous Markov chain, with transition rate matrix Q , and let T_t be the associated family of transition matrices. Then, for all $t \in \mathbb{R}_{\geq 0}$, it holds that*

$$T_t = \lim_{n \rightarrow +\infty} \left(I + \frac{t}{n} Q \right)^n .$$

One way to think about this is that, for some fixed (but large enough) $n \in \mathbb{N}$, each factor $(I + t/n Q)$ is, due to Eq. (5.7), roughly the ‘small step’ transition matrix $T_{t/n}$. The multiplication of these n terms $(I + t/n Q)^n$ is then analogous to the composition in Proposition 5.2, whereby we cover the duration t in steps of size t/n . It should be noted that this only becomes exact in the limit (as the result states), but the intuition behind it is the same regardless.

Furthermore, let us again remark that the transition-matrix representation is also convenient in that it offers an alternative representation of the conditional expectation operator:

Proposition 5.9 *Let $\{X_t\}_{t \in \mathbb{R}_{\geq 0}}$ be a continuous-time homogeneous Markov chain, with transition rate matrix Q , and let T_t be the associated family of transition matrices. Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $t \in \mathbb{R}_{\geq 0}$ and all $x \in \mathcal{X}$, it holds that $\mathbb{E}[f(X_t) | X_0 = x] = [T_t f](x)$.*

Proof Analogous to the proof of Proposition 5.3. □

Let us consider the importance of the homogeneity assumption in the preceding exposition. Indeed, it is this property that crucially allows the parametrisation to only require a single rate matrix Q . More generally, we may consider a non-homogeneous CTMC and consider the derivatives at each time point; first write the transition matrix for the interval $[s, t]$ as

$$T_s^t(x, y) \doteq P(X_t = y | X_s = x) ,$$

and differentiate to obtain

$$\left. \frac{d T_s^t}{d t} \right|_{t=s} = \lim_{t \rightarrow s^+} \frac{T_s^t - I}{t - s} =: Q_s ,$$

whence the parametrisation now requires an entire family Q_s of rate matrices—one for each point in time. Note, though, that these matrices are still transition rate matrices, in that they satisfy the properties in Definition 5.16. However, the corresponding matrix differential equation is no longer solved by a simple matrix exponential.

More generally still, for arbitrary continuous-time stochastic processes (that are neither homogeneous nor Markov) we may consider the transition rates (derivatives) not only for specific points in time but also for specific histories leading up to that time. For instance, with $\mathbf{s} = s_1, \dots, s_n$ and t in $\mathbb{R}_{\geq 0}$ and $x_s \in \mathcal{X}^n$, we may write

$$\left. \frac{d}{du} P(X_u = y \mid X_{\mathbf{s}} = x_{\mathbf{s}}, X_t = x) \right|_{u=t} =: Q_{x_{\mathbf{s}}, t}(x, y). \quad (5.8)$$

Thus, the parametrisation requires the specification of a transition rate matrix for each point in time and for each possible history before that time. It should be clear that this leads to a rather unwieldy process specification, which again goes some way in illustrating why homogeneity and Markovianity are such popular simplifying assumptions.

5.4.1 Imprecise Continuous-Time Markov Chains

With the notation and concepts for precise continuous-time stochastic processes in place, let us now turn to the imprecise generalisation. In what follows, we will consider imprecise, homogeneous continuous-time Markov chains (ICTMC). As before, we start by considering the abstract sets-of-measures definition:

Definition 5.17 (ICTMC as set of processes) An *imprecise continuous-time Markov chain* is a set \mathbb{P} of probability measures on the measurable space (Ω, \mathcal{F}) of (continuous-time) paths, with associated lower expectation operator $\underline{\mathbb{E}}_{\mathbb{P}}$ such that, for all $f \in \mathcal{L}(\mathcal{X})$ and all $s_1, \dots, s_n, t \in \mathbb{R}_{\geq 0}$ such that $s_1 < \dots < s_n < t$, it holds that

$$\underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_{s_1}, \dots, X_{s_n}] = \underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_{s_n}].$$

Furthermore, an imprecise continuous-time Markov chain is called *homogeneous* if, for all $s, t \in \mathbb{R}_{\geq 0}$, $s < t$, and all $f \in \mathcal{L}(\mathcal{X})$, it holds that $\underline{\mathbb{E}}_{\mathbb{P}}[f(X_t) \mid X_s] = \underline{\mathbb{E}}_{\mathbb{P}}[f(X_{t-s}) \mid X_0]$.

As in the discussion about imprecise discrete-time Markov chains, we distinguish between the definition by epistemic irrelevance—which is what is used above—and the definition by strong independence, which would imply that all $P \in \mathbb{P}$ are precise (homogeneous) Markov chains, and which we are explicitly not using.

Let us now consider the parametrisation of such an ICTMC. We recall that in the precise case, the canonical parameter is a single transition rate matrix Q . In

contrast, for the imprecise case, the ‘parameter’ of interest is a set \mathcal{Q} of transition rate matrices. Because a precise homogeneous CTMC is identified with a rate matrix Q , it is clear that such a set \mathcal{Q} induces a set of precise processes: simply consider all processes for which the associated rate matrix is included in \mathcal{Q} . However, this induced set then only includes homogeneous Markov processes, and, as remarked above, we aim to relax these independence assumptions. Using the parametrisation of more general precise processes, we introduce the notion of compatibility with a given set of rate matrices:

Definition 5.18 Let \mathcal{Q} be a set of transition rate matrices. Then a continuous-time stochastic process P is called *compatible* with \mathcal{Q} if, for all $\mathbf{s} = s_1, \dots, s_n$ and $t \in \mathbb{R}_{\geq 0}$ such that $s_1 < \dots < s_n < t$, and all $x_{\mathbf{s}} \in \mathcal{X}^n$, it holds that $Q_{x_{\mathbf{s}}, t} \in \mathcal{Q}$, where $Q_{x_{\mathbf{s}}, t}$ is the time- and history-dependent rate matrix associated with P , as in Eq. (5.8).

It can be verified that this definition includes, as a special case, the compatibility of homogeneous CTMCs with rate matrix Q , with a given set \mathcal{Q} , if $Q \in \mathcal{Q}$. Similarly, a non-homogeneous CTMC that is parametrised by a family Q_t is compatible with such a set if $Q_t \in \mathcal{Q}$ for all $t \in \mathbb{R}_{\geq 0}$. The ICTMC \mathbb{P} corresponding to a given set \mathcal{Q} , then, is taken to be the largest set of continuous-time stochastic processes that are compatible with this \mathcal{Q} . While perhaps not obvious, it can be proven that this set \mathbb{P} is then indeed a homogeneous ICTMC, in the sense that its corresponding lower expectations satisfy the properties of Definition 5.17.

With this ICTMC in place, let us now again consider the main inferential challenge: how to compute the corresponding lower expectation. A first attempt could be to use Propositions 5.7 and 5.9 and optimise over \mathcal{Q} ; for some fixed $f \in \mathcal{L}(\mathcal{X})$, this would give

$$\inf_{Q \in \mathcal{Q}} e^{Q_t} f.$$

If we think about what this computes, we come to the conclusion that for each $Q \in \mathcal{Q}$, there is a homogeneous CTMC for which the conditional expectation of f at time $t \in \mathbb{R}_{\geq 0}$ is indeed $e^{Q_t} f$. We therefore conclude that this computes the lower expectation with respect to all *homogeneous* CTMCs that are compatible with \mathcal{Q} . But what about the non-homogeneous and/or non-Markovian stochastic processes that we know are also included in \mathbb{P} ? It turns out that the above expression ignores their corresponding expectations and hence only yields an *upper bound* on the actual *lower* expectation. In other words, we cannot use this expression to compute the lower expectation for \mathbb{P} .

The way to proceed is analogous to the approach in Sect. 5.3.2; we first define a *local* ‘lower’ operator and then find the global lower expectation using repeated compositions of this operator through the law of iterated lower expectation. To this end, we associate with the set \mathcal{Q} the corresponding *lower transition rate operator* $\underline{Q} : \mathcal{L}(\mathcal{X}) \rightarrow \mathcal{L}(\mathcal{X})$, which is defined for all $f \in \mathcal{L}(\mathcal{X})$ and all $x \in \mathcal{X}$ as

$$[\underline{Q}f](x) \doteq \inf_{Q \in \mathcal{Q}} [Qf](x). \tag{5.9}$$

Intuitively, for small $\Delta > 0$, we can then approximate the lower expectation as

$$\mathbb{E}_{\mathbb{P}}[f(X_{\Delta}) | X_0] \approx \inf_{Q \in \mathcal{Q}} (I + \Delta Q)f = (I + \Delta \underline{Q})f,$$

where the approximation is again due to Eq. (5.7). It turns out that we can make this exact and extend the result to any time t , analogously to Proposition 5.8:

Theorem 5.5 *Let \mathcal{Q} be a non-empty set of transition rate matrices, and let \underline{Q} be the corresponding lower transition rate operator, as in Eq. (5.9). Then, for all $t \in \mathbb{R}_{\geq 0}$, there is an operator $\underline{T}_t : \mathcal{L}(\mathcal{X}) \rightarrow \mathcal{L}(\mathcal{X})$, such that*

$$\underline{T}_t = \lim_{n \rightarrow +\infty} \left(I + \frac{t}{n} \underline{Q} \right)^n.$$

These operators satisfy $\underline{T}_0 = I$, $\underline{T}_{t+s} = \underline{T}_t \underline{T}_s$ for all $t, s \in \mathbb{R}_{\geq 0}$ and $d/dt \underline{T}_t = \underline{Q} \underline{T}_t$.

Observe that this family of operators \underline{T}_t satisfies in large part the same properties as the matrix exponentials of $\underline{Q}t$ —c.f. the discussion after Proposition 5.7—with the main difference being that they are *non-linear* operators. We can now finally present the result that allows the computation of lower expectations for ICTMCs.

Theorem 5.6 *Let \mathcal{Q} be a non-empty set of transition rate matrices, with corresponding lower transition rate operator \underline{Q} , and let \mathbb{P} be the corresponding ICTMC. Suppose that \mathcal{Q} is closed, convex and has separately specified rows (i.e. is closed under recombination of the rows of its elements). Then, for all $f \in \mathcal{L}(\mathcal{X})$, all $t \in \mathbb{R}_{\geq 0}$ and all $x \in \mathcal{X}$, it holds that*

$$\mathbb{E}_{\mathbb{P}}[f(X_t) | X_0 = x] = [\underline{T}_t f](x). \tag{5.10}$$

Observe that this result needs some constraints on the rate matrix set \mathcal{Q} . This can be explained in the sense that the right-hand side of Eq. (5.10) depends, through Theorem 5.5, on the lower transition rate operator \underline{Q} . In turn, \underline{Q} depends on \mathcal{Q} through Eq. (5.9). Conversely, the left-hand side (the lower expectation) depends on the set \mathbb{P} , which in turn depends on \mathcal{Q} through the compatibility as in Definition 5.18. It turns out that for these different dependencies on \mathcal{Q} to be equivalent, we need some regularity conditions on this latter set—these are the constraints mentioned in the theorem above.

5.4.2 Limits of ICTMCs

Let us finally consider the long-term behaviour of a given homogeneous ICTMC \mathbb{P} with transition rate matrix set \mathcal{Q} and associated lower transition rate operator \underline{Q} ; we assume these to be fixed in the remainder of this section. What, then, can we say about the lower expectation of a function as time goes to infinity?

Recall that, in the discrete-time case, Theorem 5.4 established a sufficient condition for such a lower expectation to converge. This condition was *regularity* of the IDTMC. Essentially, this meant that it was possible for the IDTMC to move from any state to any other state, in exactly n steps, for some $n \in \mathbb{N}$. In the continuous-time case that we consider here, there is a similar condition: *upper reachability* between all pairs of states.

We first remark that this condition is defined using the conjugate upper transition rate operator defined as $\overline{Q}f \doteq -\underline{Q}(-f)$ for all $f \in \mathcal{L}(\mathcal{X})$. The definition of upper reachability is then analogous to that of accessibility in discrete-time but is instead defined using the transition *rates*, rather than probabilities:

Definition 5.19 Let \mathbb{P} be an ICTMC with associated upper transition rate operator \overline{Q} , as defined above. For any two states $x, y \in \mathcal{X}$, y is said to be *upper reachable* from x , if there is a sequence $x_0, \dots, x_n \in \mathcal{X}$, $n \in \mathbb{N}$, such that $x_0 = x$, $x_n = y$ and, for all $i \in \{1, \dots, n\}$, it holds that $x_i \neq x_{i-1}$ and $[\overline{Q} \mathbb{I}_{x_i}](x_{i-1}) > 0$.

Let us in particular consider the final condition in this definition. From the conjugacy between the lower and upper transition rate operators, and the definition of the former, we can rewrite this requirement as saying that

$$0 < [\overline{Q} \mathbb{I}_{x_i}](x_{i-1}) = \sup_{Q \in \mathcal{Q}} [Q \mathbb{I}_{x_i}](x_{i-1}) = \sup_{Q \in \mathcal{Q}} Q(x_{i-1}, x_i).$$

Thus, upper reachability of y , from x , requires that there exists a sequence of states from x to y such that, at each step in this sequence, there is *some* transition rate matrix $Q \in \mathcal{Q}$ which assigns strictly positive ‘speed’ of moving from the current state in this sequence, to the next one. In other words, it should be possible for these transitions to happen according to some of the models in our set \mathbb{P} , but not necessarily all, and there can be a different model allowing for this possibility at each step. This can now be used to state the following result:

Theorem 5.7 *Let \mathbb{P} be an ICTMC and suppose that, for all $x, y \in \mathcal{X}$, y is upper reachable from x . Then, there is a unique lower expectation operator $\mathbb{E}_{\mathbb{P}}[\cdot(X_{+\infty})] : \mathcal{L}(\mathcal{X}) \rightarrow \mathbb{R}$ such that, for all $f \in \mathcal{L}(\mathcal{X})$ and all $x \in \mathcal{X}$,*

$$\mathbb{E}_{\mathbb{P}}[f(X_{+\infty})] = \lim_{t \rightarrow +\infty} \mathbb{E}_{\mathbb{P}}[f(X_t) \mid X_0 = x] = \lim_{t \rightarrow +\infty} [T_t f](x).$$

5.5 Literature and Further Reading

Let us conclude this chapter by providing pointers to the literature on which the material in this chapter is based. We will also briefly discuss some parts of the literature that are related but not quite the same as what we covered here.

First of all, there exists an extensive body of literature on (precise) Markov chains, both in discrete- and in continuous times. It would be nigh impossible to give a complete overview here, but we think that [1, 38] make excellent introductory reads. For a broad and general introduction to the theory for imprecise probability, which lies at the heart of the models that we discussed here, we refer the reader to [3, 50]. The difference between the notions of strong independence and epistemic irrelevance—which we have stressed repeatedly and which is a crucial property of imprecise Markov chains as we treated them here—is discussed, e.g. in [5, 37].

For the interpretation of Markov chains using probability trees, see, for example, [13, 18, 32]. This interpretation is also closely related to the *game-theoretic* formalisation of probabilities using the theory of martingales (which we did not cover here). The interested reader may want to pursue [13, 32, 49].

For an account of the general theory of Bayesian networks, see [39]. For their imprecise generalisation—credal networks—references [2, 6, 7, 9, 11] discuss a lot of the general theory.

Imprecise *discrete*-time Markov chains are discussed, e.g. in [15, 17, 27]. For imprecise *continuous*-time Markov chains, see [30, 44]. A treatment of the matrix exponential, which is crucial to computational methods for CTMCs, is given in [48]. Reference [19] discusses the current state-of-the-art to efficiently compute the imprecise generalisation of the matrix exponential, which we have seen is crucial for computing inferences in ICTMCs.

Detailed treatments on the long-term (limit) behaviour in IDTMCs can be found in [14, 16, 26, 45]. Reference [10] provides the necessary and sufficient conditions for the limit behaviour of ICTMCs, and [19] also discusses computational methods to numerically approximate this limit. We remark that Theorems 5.4 and 5.7 in this chapter are stated in a simplified form compared to their statement in the literature. In particular, the results in [10, 14] are stronger; for instance, [10] in fact provides *necessary* and sufficient conditions for the convergence of an ICTMC, whereas Theorem 5.7 only states a sufficient condition.

Some examples of the merits of imprecise Markov chains in applications are provided by [40, 46, 47]. A domain for which the applicability of (imprecise) Markov chains has been extensively studied, is queueing theory [8, 33–36].

A generalisation of Markov chains that we have not discussed, but which is nevertheless important in many practical applications, is *hidden* Markov chains. There, the stochastic process cannot be observed directly, but only through a noisy measurement model. Their imprecise treatment is discussed, e.g. in [4, 12, 31].

Fields that are closely related to the theory of imprecise Markov chains are controlled Markov processes [21] and Markov decision processes [22, 28, 41, 51]. There also, the process under study has its parameters changed over time. However,

the goal there is not to represent uncertainty and change these parameters to compute robust bounds on quantities of interest. Rather, their aim is to optimise the process evolution towards some operational target.

Finally, we again emphasise that our treatment uses epistemic irrelevance, which we distinguish from using strong independence. There is, however, an extended body of literature also on the latter. These alternative models are known as Markov chains under strong independence, e.g. in [27], as interval Markov chains [20, 29, 42, 43] or as Markov set chains [23–25].

Acknowledgments The author wishes to express his sincere gratitude to Gert de Cooman and Jasper De Bock, for their helpful comments and suggestions during the writing of this chapter. He also wants to thank the reviewer, whose comments and suggestions further helped to improve this work.

References

1. W.J. Anderson, *Continuous-Time Markov Chains, An Applications-Oriented Approach*. Springer Series in Statistics (Springer, New York, 1991)
2. A. Antonucci, C.P. de Campos, M. Zaffalon, Probabilistic graphical models, in *Introduction to Imprecise Probabilities*, ed. by T. Augustin, F.P.A. Coolen, G. De Cooman, M.C.M. Troffaes (Wiley, New York, 2014)
3. T. Augustin, F.P.A. Coolen, G. De Cooman, M.C.M. Troffaes, *Introduction to Imprecise Probabilities* (Wiley, New York, 2014)
4. A. Benavoli, M. Zaffalon, E. Miranda, Robust filtering through coherent lower previsions. *IEEE Trans. Autom. Control* **56**(7), 1567–1581 (2011)
5. I. Couso, S. Moral, P. Walley, A survey of concepts of independence for imprecise probabilities. *Risk Decis. Policy* **5**(2), 165–181 (2000)
6. F.G. Cozman, Credal networks. *Artif. Intell.* **120**, 199–233 (2000)
7. F.G. Cozman, Graphical models for imprecise probabilities. *Int. J. Approx. Reason.* **39**(2–3), 167–184 (2005)
8. R.J. Crossman, P. Coolen-Schrijner, F.P. Coolen, Time-homogeneous birth-death processes with probability intervals and absorbing state. *J. Stat. Theory Pract.* **3**(1), 103–118 (2009)
9. J. De Bock, Credal networks under epistemic irrelevance: theory and algorithms. Ph.D. thesis, Ghent University (2015)
10. J. De Bock, The limit behaviour of imprecise continuous-time Markov chains. *J. Nonlinear Sci.* **27**(1), 159–196 (2017)
11. J. De Bock, Credal networks under epistemic irrelevance. *Int. J. Approx. Reason.* **85**, 107–138 (2017)
12. J. De Bock, G. De Cooman, An efficient algorithm for estimating state sequences in imprecise hidden Markov models. *J. Artif. Intell. Res.* **50**, 189–233 (2014)
13. G. De Cooman, F. Hermans, Imprecise probability trees: Bridging two theories of imprecise probability. *Artificial Intelligence* **172**(11), 1400–1427 (2008)
14. G. De Cooman, F. Hermans, E. Quaeghebeur, Imprecise Markov chains and their limit behavior. *Probab. Eng. Inf. Sci.* **23**(4), 597–635 (2009)
15. G. De Cooman, F. Hermans, A. Antonucci, M. Zaffalon, Epistemic irrelevance in credal nets: the case of imprecise Markov trees. *Int. J. Approx. Reason.* **51**(9), 1029–1052 (2010)
16. G. De Cooman, J. De Bock, S. Lopatzidis, A pointwise ergodic theorem for imprecise Markov chains, in *Proceedings of ISIPTA 2015*, pp. 107–115 (2015)

17. G. De Cooman, J. De Bock, S. Lopatzidis, Imprecise stochastic processes in discrete time: global models, imprecise Markov chains, and ergodic theorems. *Int. J. Approx. Reason.* **76**, 18–46 (2016)
18. S. Destercke, G. De Cooman, Relating epistemic irrelevance to event trees. *Soft Methods for Handling Variability and Imprecision* (Springer, New York, 2008), pp. 66–73
19. A. Erreygers, J. De Bock, Imprecise continuous-time Markov chains: Efficient computational methods with guaranteed error bounds, in *Proceedings of ISIPTA 2017*, pp. 145–156 (2017)
20. S. Galdino, Interval continuous-time Markov chains simulation, in *Proceedings of the 2013 International Conference on Fuzzy Theory and Its Applications*, pp. 273–278 (2013)
21. X. Guo, O. Hernández-Lerma, Continuous-time controlled Markov chains. *Ann. Appl. Probab.* **13**(1), 363–388 (2003)
22. X. Guo, O. Hernández-Lerma, *Continuous-Time Markov Decision Processes* (Springer, New York, 2009)
23. D.J. Hartfiel, Sequential limits in Markov set-chains. *J. Appl. Probab.* **28**(4), 910–913 (1991)
24. D.J. Hartfiel, *Markov Set-Chains*. Lecture Notes in Mathematics, vol. 1695 (Springer, New York, 1998)
25. D.J. Hartfiel, E. Seneta, On the theory of Markov set-chains. *Adv. Appl. Probab.* **26**, 947–964 (1994)
26. F. Hermans, G. De Cooman, Characterisation of ergodic upper transition operators. *Int. J. Approx. Reason.* **53**(4), 573–583 (2012)
27. F. Hermans, D. Škulj, Stochastic processes, in *Introduction to Imprecise Probabilities*, ed. by T. Augustin, F.P.A. Coolen, G. De Cooman, M.C.M. Troffaes (Wiley, New York, 2014)
28. H. Itoh, K. Nakamura, Partially observable Markov decision processes with imprecise parameters. *Artificial Intelligence* **171**, 453–490 (2007)
29. I.O. Kozine, L.V. Utkin, Interval-valued finite Markov chains. *Reliable Computing* **8**(2), 97–113 (2002)
30. T. Krak, J. De Bock, A. Siebes, Imprecise continuous-time Markov chains. *Int. J. Approx. Reason.* **88**, 452–528 (2017)
31. T. Krak, J. De Bock, A. Siebes, Efficient computation of updated lower expectations for imprecise continuous-time hidden Markov chains, in *Proceedings of ISIPTA 2017*, pp. 193–204 (2017)
32. S. Lopatzidis, Robust modelling and optimisation in stochastic processes using imprecise probabilities, with an application to queueing theory. Ph.D. thesis, Ghent University (2017)
33. S. Lopatzidis, J. De Bock, G. De Cooman, Calculating bounds on expected return and first passage times in finite-state imprecise birth-death chains, in *Proceedings of ISIPTA 2015*, pp. 177–186 (2015)
34. S. Lopatzidis, J. De Bock, G. De Cooman, Computational methods for imprecise continuous-time birth-death processes: a preliminary study of flipping times, in *Proceedings of ISIPTA 2015*, p. 344 (2015)
35. S. Lopatzidis, J. De Bock, G. De Cooman, S. De Vuyst, J. Walraevens, Robust queueing theory: an initial study using imprecise probabilities. *Queueing Systems* **82**(1–2), 75–101 (2016)
36. S. Lopatzidis, J. De Bock, G. De Cooman, Computing lower and upper expected first passage and return times in imprecise birth-death chains. *Int. J. Approx. Reason.* **80**, 137–173 (2017)
37. E. Miranda, G. De Cooman, Structural judgements, in *Introduction to Imprecise Probabilities*, ed. by T. Augustin, F.P.A. Coolen, G. De Cooman, M.C.M. Troffaes (Wiley, New York, 2014)
38. J.R. Norris, *Markov Chains* (Cambridge University Press, Cambridge, 1998)
39. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, 1988)
40. C. Rottondi, A. Erreygers, G. Verticale, J. De Bock, Modelling spectrum assignment in a two-service flexi-grid optical link with imprecise continuous-time Markov chains, in *Proceedings of DRCN 2017*, pp. 39–46 (2017)
41. J.K. Satia, R.E. Lave, Markovian decision processes with uncertain transition probabilities. *Operations Research* **21**, 728–740 (1973)

42. D. Škulj, Finite discrete time Markov chains with interval probabilities, in *Soft Methods for Integrated Uncertainty Modelling*, ed. by J. Lawry, E. Miranda, A. Bugarin, S. Li, M.A. Gil, P. Grzegorzewski, O. Hryniewicz (Springer, New York, 2006), pp. 299–306
43. D. Škulj, Regular finite Markov chains with interval probabilities, in *Proceedings of ISIPTA 2007*, pp. 405–413 (2007)
44. D. Škulj, Efficient computation of the bounds of continuous time imprecise Markov chains. *Appl. Math. Comput.* **250**(C), 165–180 (2015)
45. D. Škulj, R. Hable, Coefficients of ergodicity for Markov chains with uncertain parameters. *Metrika* **76**(1), 107–133 (2013)
46. Y. Soullard, A. Antonucci, S. Destercke, Technical gestures recognition by set-valued hidden Markov models with prior knowledge, in *Soft Methods for Data Science*, pp. 455–462 (2017)
47. M. Troffaes, J. Gledhill, D. Škulj, S. Blake, Using imprecise continuous time Markov chains for assessing the reliability of power networks with common cause failure and non-immediate repair, in *Proceedings of ISIPTA 2015*, pp. 287–294 (2015)
48. C.F. Van Loan, A Study of the Matrix Exponential, Numerical Analysis Report No. 10, University of Manchester, Manchester, UK, August 1975, Reissued as MIMS EPrint 2006.397, Manchester Institute for Mathematical Sciences, The University of Manchester, UK
49. V. Vovk, G. Shafer, Game-theoretic probability, in *Introduction to Imprecise Probabilities*, ed. by T. Augustin, F.P.A. Coolen, G. De Cooman, M.C.M. Troffaes, (Wiley, New York, 2014)
50. P. Walley, *Statistical Reasoning with Imprecise Probabilities* (Chapman and Hall, London, 1991)
51. C.C. White, H.K. Eldeib, Markov decision-processes with imprecise transition-probabilities. *Operations Research* **42**, 739–749 (1994)

Chapter 6

Fundamentals of Filtering



Cristian Greco and Massimiliano Vasile

Abstract Accurately estimating the state of a dynamical system is of fundamental importance in a variety of applications, from engineering challenges to everyday life. This task is complex because uncertainties typically affect the dynamical behaviour as well as the available observations of the (hidden) state. The goal of this chapter is to provide a comprehensive overview, from the probabilistic problem statement to methods for its solution. In particular the focus will be on filtering problems for time-continuous state evolution equations and time-discrete observations. It will be shown that, except for very few cases, the filtering problem has no closed-form solution, which is generally infinite-dimensional. Hence, several practical algorithms to find an approximate solution are presented.

Keywords State estimation · Uncertainty propagation · Inference · Navigation · Filtering algorithms

6.1 The State Estimation Problem

Filtering theory addresses the problem of estimating the state of a system given an uncertain knowledge of its dynamical equations and noisy indirect observations. Being a mathematical branch of the general stochastic processes theory, it finds wide application in several fields spanning from engineering to physics, from biology to medical sciences. Among the most notable examples of application there are the *Global Positioning System* (GPS) technology, *brain imaging* methods, *signal processing* techniques, the prediction of the evolution of (potential) *infectious diseases* or *environmental catastrophes* and any other system modelled by stochastic differential equations [51].

Modern filtering techniques combine the dynamical and measurement information in order to optimally estimate a system state or some model parameters, where

C. Greco (✉) · M. Vasile
University of Strathclyde, Glasgow, UK
e-mail: c.greco@strath.ac.uk; massimiliano.vasile@strath.ac.uk

the concept of *optimality* will be formalized in Sect. 6.1.3. This process is nothing more than an inversion problem of the measurements equations $\mathbf{y} = \mathbf{h}(t, \mathbf{x}) + \boldsymbol{\epsilon}$ —where \mathbf{y} and \mathbf{x} are, respectively, the observation and state vector, \mathbf{h} is the observation model function and $\boldsymbol{\epsilon}$ is the measurement error—taking into account an initial estimate of the state and the equations governing the state evolution. Historically, this step has been tackled from two conceptually different perspectives:

- *Probabilistic approach*: the errors in the dynamics and the observations are characterized as random variables with known probability distributions. Hence, the goal is to compute the conditional probability $p(\mathbf{x}_k | \mathbf{y}_{1:k})$, whose knowledge represents the complete solution to the filtering problem. Indeed, in the general nonlinear case, the filter state is infinite-dimensional and represented by the conditional density function [29]. Assumptions need to be made on the distributions of random variables \mathbf{x} and \mathbf{y} to obtain a finite-dimensional solution, as will be shown in Sect. 6.3. Out of the conditional probability density function, common choices for the optimal estimate of the state are the distribution mean, median, mode and so on.
- *Statistical approach*: the errors in the dynamics and the observations are considered as unknown but deterministic; the goal is then to minimise a chosen performance index which is a function of the deviation between the obtained measurements and the computed observations (through the deterministic part of the observation model). The dynamical equations are then formulated as constraints of the minimisation procedure. A well-known statistical approach for which this optimisation step has a closed-form solution is the (possibly recursive) weighted *least squares* method [58] which minimizes the sum of the weighted square deviations. In general, different performance indices result in different estimates of the state.

In general, this chapter pursues the probabilistic approach to state estimation in its theoretical development. However, when looking at the practical implementation of filtering methods, often this boundary starts to fade. This results from the different possible derivations and interpretations of the same method. As an example, the time-discrete linear Kalman filter (see Sect. 6.3.1) can be derived starting either from a probabilistic reasoning [29] or from a statistical approach [58], leading to the same final algorithm.

Often, engineering books focus on the latter approach as it gives a simpler high-level interpretation to the filtering step, i.e. as a constrained optimisation. Moreover, it requires little or no knowledge of probability and allows for a direct focus on the filter implementation. On the other hand, the probabilistic approach provides solid mathematical justification to each assumption and step at the cost of increased theoretical complexity. Indeed, the knowledge of probability theory is essential and an understanding of stochastic differential equations often useful. In this chapter the latter approach will be pursued to provide a comprehensive awareness of filtering theory and the advantages and approximations resulting from a selected filter. Nonetheless, the following development will attempt to simplify the mathematically formal description retaining the minimum set of fundamental

concepts needed to have a comprehensive understanding of filtering theory. For a more formal mathematical treatment, the reader should refer to the classical work from Jazwinski [29].

The remainder of the chapter is structured as follows. Section 6.1 will present the framework of state estimation problems for time-continuous systems and introduce the needed working mathematical concepts. As the probabilistic approach is pursued, Sect. 6.2 will discuss the exact and approximated methods for propagating probability distributions through generic nonlinear transformations, which in this setting are the dynamical equations and the observation model. Finally, Sect. 6.3 introduces several practical algorithms to compute the (often approximated) solution of the filtering problem, each more suitable under different working conditions and assumptions. To summarise the chapter, Sect. 6.4 provides a final overview of the presented topics.

6.1.1 Building Blocks

Generally in aerospace applications, ‘filtering’ and ‘state estimation’ are interchangeable to indicate the process of propagating the state distribution knowledge through a dynamical model, and updating this estimate when new observations are available, potentially decreasing (filtering) the noise contributions. On the other hand, in mathematical applications, the filtering problem defines only the update step, when prior information and noisy measurements are combined. For the notation employed in this book, a filter is a model handling both the propagation and the update; therefore, the two aforementioned names shall be considered synonyms.

Modern filtering theory employs a general framework to handle a great variety of mathematical problems. A requirement of the system to be filtered is to be a stochastic process with possibly hidden (unobserved) states in the most general cases. In addition, some a priori estimate of the initial state should be available. Therefore, three main elements are necessary components to define a filter:

- *Dynamical model*: a set of equations mapping the state at time t_i to the state at time t_j . Generally in the continuous-time case, the dynamical model is described by a set of differential equations. In line with the applications outlined, the conventional set of dynamical equations will be described as first-order ordinary differential equations in the state space form:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) + G(t, \mathbf{x})\mathbf{w}, \quad (6.1)$$

where \mathbf{f} is the deterministic system dynamics, \mathbf{w} is a white Gaussian noise and G is its coefficient matrix. Usually, the stochastic term is introduced to model possible approximation errors. It is worth clarifying that with this formulation we intend to encompass the full range of dynamical models and we do not restrict to natural dynamics only. For example, while control forces \mathbf{u} are usually included in this notation writing $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u})$, in this chapter the explicit dependency

will be hidden, without introducing any restriction, for the sake of clarity. In state estimation, the interest often lies in the computation of conditional probability distributions on the propagated state. Section 6.2 will introduce several approximate methods employed to numerically compute this distribution from the general dynamics in Eq. (6.1).

- *Observation model*: a set of nonlinear equations describing how a state is mapped to the measurements taking into account a noise term ϵ modelling the error:

$$\mathbf{y} = \mathbf{h}(t, \mathbf{x}) + \epsilon . \tag{6.2}$$

In this chapter, the noise is considered to have zero-mean and to be additive for simplicity, whereas similar derivations can be achieved with non-additive noise [51].

The deterministic term is employed to compute the error-free *predicted observations* (or often called *computed observations*), an important concept both in the statistical perspective and in practical algorithm formulation [58]. Real observations are naturally subject to noise and biases dependent on the particular type of measurement, which should be modelled as well to provide the filter with more information on the observation received.

- *Initial distribution*: the a priori knowledge of the state probability distribution at the initial time, generally assumed independent of any dynamical or observational noise:

$$p(\mathbf{x}_0) . \tag{6.3}$$

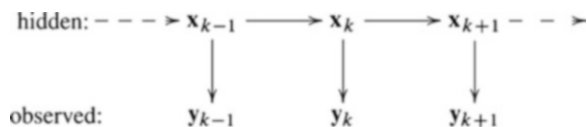
Once these components are defined, the goal of the estimation process is to compute the optimal combination of generally conflicting information from dynamical knowledge and received observations (Fig. 6.1).

This optimal combination—called inference in the Bayesian approach—is mathematically formulated as a nonlinear inverse problem. Let’s assume we have a time-ordered measurement vector $\mathbf{y}_{1:l} = [\mathbf{y}_1(\mathbf{x}_1), \dots, \mathbf{y}_k(\mathbf{x}_k), \dots, \mathbf{y}_l(\mathbf{x}_l)]$. The complete solution of the state estimation problem is given by the state joint probability distribution conditional on all the observations:

$$p(\mathbf{x}_{0:T} | \mathbf{y}_{1:l}) , \tag{6.4}$$

where l is the number of observations and T is the number of times at which the state should be estimated. This state distribution captures all the statistical information provided both by the measurements and the prior state knowledge [29].

Fig. 6.1 Scheme of hidden dynamical system with discrete observations [51]



However, this posterior is expensive to compute as it is a joint distribution over all \mathbf{x}_k , conditionally dependent on all the observations $\mathbf{y}_{1:t}$. Furthermore, in real-time scenarios, every time a new observation is available, the full joint posterior distribution needs to be computed again.

6.1.1.1 State Marginalization

The major computational complexity in Eq. (6.4) stems from the joint nature of the posterior distribution. Indeed, whenever a new observation is available for a different time step, the posterior distribution's dimensionality will increase, degrading the computational efficiency severely. If the main interest concerns the computation of the state estimate \mathbf{x}_k at a specific time step, like in real-time applications, this excessive numerical burden can be reduced by computing the marginal distribution instead:

$$p(\mathbf{x}_k | \mathbf{y}_{1:t}) . \quad (6.5)$$

This approach reduces dramatically the posterior dimensionality and, therefore, improves the computational performance.

6.1.1.2 Markov and Independence Assumptions

Given the knowledge of \mathbf{x}_k , a Markov process links the future process probability law for $t > t_k$ only to \mathbf{x}_k , independently of how this state was reached. This is analogous to an ordinary differential equation which links the state rate $\dot{\mathbf{x}}_k$ only to simultaneous time t_k , state \mathbf{x}_k (and possibly parameters) [29]. If we restrict to this nevertheless wide class of models, the system dynamics in Eq. (6.1) is a Markov process and the time-ordered collections of states $\mathbf{x}_{1:T}$ a Markov sequence. More formally, this class respects the Markov property which simplifies the state dependency:

$$p(\mathbf{x}_k | \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) . \quad (6.6)$$

Also measurements can be included in the Markov model. Then, the previous property can be generalised as [51]:

$$p(\mathbf{x}_k | \mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k | \mathbf{x}_{k-1}) . \quad (6.7)$$

As a consequence of these two assumptions, \mathbf{x}_k does not depend on anything which happened before the time t_{k-1} given \mathbf{x}_{k-1} .

Another traditional assumption is to consider the measurement \mathbf{y}_k to be conditionally independent of previous state history or observations:

$$p(\mathbf{y}_k | \mathbf{x}_{0:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k | \mathbf{x}_k). \quad (6.8)$$

In the next sections, these assumptions will be key in further reducing the complexity of the computation of joint probability distributions.

6.1.2 Filtering Problem Formulation

The estimation problem can be divided in different categories according to which subset of measurements is considered in the computation of the marginal conditional posterior distribution in Eq. (6.5).

Depending on the application, the state to be estimated \mathbf{x}_k could be at a previous, contemporary or later time step than the time of the last observation \mathbf{y}_l . Hence, for each scenario, different observations should be taken into account in the computation of the conditional distribution in Eq. (6.5).

The estimation problem is called *smoothing* when $t_k < t_l$. As an example, this problem is faced in post-processing applications, when all the measurements in time are available, and the interest is computing the best estimate possible of the state using also later observations. The marginal posterior distribution is $p(\mathbf{x}_k | \mathbf{y}_{1:l})$.

The estimation problem is labelled *filtering* when $t_k = t_l$. This case is typical of real-time applications, when the state estimate is to be updated after a new observation is available. To distinguish the notation, the marginal posterior distribution can be written as $p(\mathbf{x}_k | \mathbf{y}_{1:k})$.

Lastly, if $t_k > t_l$, the estimation problem is called *prediction*. The dynamical information is used to predict the state distribution at times after the last observation. The marginal distribution to be computed is $p(\mathbf{x}_{k+\Delta} | \mathbf{y}_{1:k})$, where Δ indicates a future time step after t_k .

The main focus of this chapter is on real-time applications; therefore, the *filtering* framework will be presented together with filter algorithms. On the other hand, most of the theoretical and algorithmic notions that will be introduced have shared cores with *smoothing* and *prediction*. Therefore, the concepts presented in this chapter are easily transferable to the other two problem scenarios.

6.1.3 Bayesian Approach for Filtering

For filtering applications, the full joint posterior distribution $p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})$ in Eq. (6.4) can be computed by Bayesian inference:

$$p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k} | \mathbf{x}_{0:k}) p(\mathbf{x}_{0:k})}{p(\mathbf{y}_{1:k})}. \quad (6.9)$$

With the Markov assumption, the terms in the numerator of the right-hand side have convenient properties which simplify the dependencies. Indeed, the joint prior distribution on the states $p(\mathbf{x}_{0:k})$ is simplified as:

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_0) \prod_{j=1}^k p(\mathbf{x}_j | \mathbf{x}_{0:j-1}) = p(\mathbf{x}_0) \prod_{j=1}^k p(\mathbf{x}_j | \mathbf{x}_{j-1}), \quad (6.10)$$

where the first identity comes from the definition of joint probability distribution and the second from the Markov property in Eq. (6.6).

In addition, the joint probability on observations can be manipulated as:

$$p(\mathbf{y}_{1:k} | \mathbf{x}_{0:k}) = \prod_{j=1}^k p(\mathbf{y}_j | \mathbf{x}_{0:k}) = \prod_{j=1}^k p(\mathbf{y}_j | \mathbf{x}_j). \quad (6.11)$$

The first identity stems from the measurements' independence as a result of the functional relationship in Eq. (6.2) and the random nature of the associated noise, while the second identity comes from the conditional independence in Eq. (6.8).

The denominator of Eq. (6.9) has no dependency on the state to be estimated. Therefore, it can be seen as a normalization factor, which is therefore possible to discard in some algorithmic implementations [51]. All these simplifications result in Eq. (6.9) to be reformulated as:

$$p(\mathbf{x}_{1:k} | \mathbf{y}_{1:k}) \propto \prod_{j=1}^k p(\mathbf{y}_j | \mathbf{x}_j) \cdot p(\mathbf{x}_0) \prod_{j=1}^k p(\mathbf{x}_j | \mathbf{x}_{j-1}). \quad (6.12)$$

However, the computation of this joint posterior is still numerically demanding. As introduced in Sect. 6.1.1.1, if we shift the goal on finding the marginal probability distribution, the complexity of the distribution reduces dramatically. Therefore, with this restriction, the posterior to be computed reduces to:

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_{1:k} | \mathbf{x}_k) \cdot p(\mathbf{x}_k)}{p(\mathbf{y}_{1:k})}. \quad (6.13)$$

A different formulation to compute the probability in Eq. (6.13), which will prove key in the next section, is obtained by applying Bayes rule only with respect to the last observation:

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{y}_{1:k}) &= p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{y}_k) \\ &= \frac{p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_{1:k-1}) \cdot p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})} \\ &= \frac{p(\mathbf{y}_k | \mathbf{x}_k) \cdot p(\mathbf{x}_k | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}, \end{aligned} \quad (6.14)$$

where the last identity stems from the conditional independence of the observations again.

It is straightforward to see how the latter suits a sequential formulation, where the prior $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ of the current step is computed by propagating the posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ of the previous one

The first term in the right-hand side numerator is the conditional probability of the measurements \mathbf{y}_k given the state \mathbf{x}_k , at time step t_k . By recalling the observation measurements in Eq. (6.2) and dropping the explicit time dependence, this probability can be written as:

$$p(\mathbf{y}_k|\mathbf{x}_k) = p(\mathbf{h}(\mathbf{x}_k) + \boldsymbol{\epsilon}_k|\mathbf{x}_k) = p_\epsilon(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k)) , \quad (6.15)$$

where the latter term is the density of the error evaluated at $\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k)$. Intuitively, this result states that for a given state \mathbf{x}_k , the probability of receiving a specific measurement only depends on the discrepancy between the modelled observation $\mathbf{h}(\mathbf{x}_k)$ and the received one \mathbf{y}_k . From the relation in Eq. (6.15), it is straightforward to derive useful relations for the posterior distribution moments, which could also serve as an argument for the latter equality (for a formal proof, see Jazwinski [29]). The conditional expectation is simply given by the computed observations:

$$E\{\mathbf{y}_k|\mathbf{x}_k\} = E\{\mathbf{h}(\mathbf{x}_k)|\mathbf{x}_k\} + E\{\boldsymbol{\epsilon}_k|\mathbf{x}_k\} = \mathbf{h}(\mathbf{x}_k) , \quad (6.16)$$

where the first equality results from the linearity of the expectation operator, whereas the second comes from a well-known property of conditional expectations $E\{\mathbf{f}(\mathbf{x})|\mathbf{x}\} = \mathbf{f}(\mathbf{x})$ (see Theorem 2.9 of Jazwinski [29]) and from the null mean of the white noise. The second-order central moment can be derived as:

$$\begin{aligned} E\left\{(\mathbf{y}_k - E\{\mathbf{y}_k|\mathbf{x}_k\})(\mathbf{y}_k - E\{\mathbf{y}_k|\mathbf{x}_k\})^T|\mathbf{x}_k\right\} &= \\ E\left\{(\mathbf{h}(\mathbf{x}_k) + \boldsymbol{\epsilon}_k - \mathbf{h}(\mathbf{x}_k))(\mathbf{h}(\mathbf{x}_k) + \boldsymbol{\epsilon}_k - \mathbf{h}(\mathbf{x}_k))^T|\mathbf{x}_k\right\} &= \\ E\left\{\boldsymbol{\epsilon}_k\boldsymbol{\epsilon}_k^T\right\} &= R_k , \end{aligned} \quad (6.17)$$

where the conditioning on \mathbf{x}_k dropped because of the measurement error independence from the state. Similarly, it can be easily shown that higher-order central moments of this posterior distribution coincide with the same moments of the distribution on $\boldsymbol{\epsilon}_k$.

The denominator of Eq. (6.14) can be computed with the law of total probability [54]:

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k . \quad (6.18)$$

Therefore, Eq. (6.14) can be written as:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k) \cdot p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k}. \quad (6.19)$$

It has been shown above that the first term in the integral is computed by using the observation model and the associated error probability function.

However, a tool to propagate the conditional probability in the time interval between successive measurements, i.e. from $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ at t_{k-1} to $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ at t_k , is still needed. To this end, the differential equations governing the conditional probability evolution will now be introduced.

6.1.3.1 Conditional Probability Evolution Between Observations

The missing bit of information to compute the posterior distribution in Eq. (6.19) is how to obtain the new prior $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$. This distribution characterises how the state at time t_k is influenced by previous observations, from t_{k-1} backwards. For its computation, we suppose that the previous step in the sequential filtering scheme has been solved, and therefore the probability $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ is known. The goal of this section is to present tools for the propagation of this conditional density from t_{k-1} to t_k when no new observations are received:

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \rightarrow p(\mathbf{x}_k|\mathbf{y}_{1:k-1}). \quad (6.20)$$

In the framework of Markov processes generated by stochastic differential equations like Eq. (6.1), Kolmogorov derived equations for the exact evolution of the density function $p(t, \mathbf{x})$, characterising the process state, and for the process transition density $p(t, \mathbf{x}_t|\tau, \mathbf{x}_\tau)$, characterising the process state evolution. The *Kolmogorov forward equation*, also known as *Fokker-Planck* or *Kolmogorov-Fokker-Planck*, is a partial differential equation that, for Markov diffusion processes, is given by Challa and Faruqi [13] and Risken [47]:

$$\frac{\partial p}{\partial t} = - \sum_i \frac{\partial p f_i}{\partial x_i} + \frac{1}{2} \sum_i \sum_j \frac{\partial^2 p (G Q G^T)_{ij}}{\partial x_i \partial x_j}, \quad (6.21)$$

where Q is the stochastic noise process covariance and the dependencies have not been explicitly written. It goes without saying that Eq. (6.21) holds under existence and continuity assumptions on the involved partial derivatives. It is worth underlining again that the equation above holds for both $p(t, \mathbf{x})$ and $p(t, \mathbf{x}|\tau, \mathbf{x})$. This equation has a closed-form solution in a limited number of simplified cases [6]. Nonetheless, Kolmogorov equation remains a powerful tool for theoretical development, as well as nonlinear filtering techniques which solve it by numerical approaches [12–15, 53]. This equation has been generalised to include more general stochastic perturbations other than the Gaussian white noise [50], and a corresponding estimation algorithm based on perturbation theory has been developed [40].

One important consequence of the Kolmogorov equation is the possibility to write down equations of motion for the density moments. The first two moments' evolution is described by Challa and Faruqi [13]:

$$\frac{dE\{\mathbf{x}_t\}}{dt} = E\{\mathbf{f}(t, \mathbf{x}_t)\} \quad (6.22)$$

$$\begin{aligned} \frac{dP}{dt} = & \left(E\left\{\mathbf{x}_t \mathbf{f}^T(t, \mathbf{x}_t)\right\} - E\{\mathbf{x}_t\}E\{\mathbf{f}(t, \mathbf{x}_t)\}^T \right) \\ & + \left(E\left\{\mathbf{f}(t, \mathbf{x}_t) \mathbf{x}_t^T\right\} - E\{\mathbf{f}(t, \mathbf{x}_t)\}E\{\mathbf{x}_t\}^T \right) + E\left\{GQG^T\right\}, \end{aligned} \quad (6.23)$$

where $P(t) = E\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^T\}$ is the covariance matrix at time t . In the general nonlinear case, these equations are not ordinary differential equations and involve dependencies on higher-order moments through the expectation operator. However, these equations could be simplified if approximations on the probability distribution are introduced, leading to important schemes for numerical algorithms.

In the absence of new observations, i.e. between two measurements times, the evolution of the conditional density $p(t, \mathbf{x}|\mathbf{y})$ equals the prior density $p(t, x)$ [29]. Hence, Kolmogorov forward equation can be used to propagate directly the conditional probability (see Eq. (6.20)). Equivalently, this density can be computed by the *Chapman-Kolmogorov equation* which links the conditional probabilities at t_{k-1} and t_k through the transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ [51]:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}, \end{aligned} \quad (6.24)$$

where the first equality follows from the definition of marginal densities, the second equality stems from the definition of the joint probability with respect to conditional one and the latter comes from the Markov property in Eq. (6.7). As already stated, the process transition density evolution is described by Kolmogorov equation.

The exact nonlinear Bayesian Filtering description of the marginal conditional density is now complete. The Filtering equations can be summarised as follows:

- *Between observations:*
 $p(t, \mathbf{x}|\mathbf{y})$ evolves according to the Kolmogorov Equation (6.21) or Chapman-Kolmogorov Equation (6.24).
- *At an observation:*
 $p(t, \mathbf{x}|\mathbf{y})$ is updated by Bayes' rule according to Eq. (6.19).

Section 6.2 will describe practical methods for the propagation of the density distribution through the dynamical system and the update step.

6.1.4 Batch Processor vs. Sequential Filtering

As seen in the previous section, one alternative in the inverse problem solution is to consider a set of observations $\mathbf{y}_{1:k}$ at once instead of sequentially (see Eq. (6.13)). This approach, called *batch processor*, employs the dynamical model to map observations at different times to t_k , usually by means of the state transition matrix and linearized observation models [58]. In smoothing applications, one advantage of this procedure is that all the available information is exploited, also the knowledge coming from possible measurements later in time. On the other hand, the batch processor becomes intractable when the number of observations becomes sufficiently high, leading to a high-dimensional and highly overdetermined inversion problem if we consider all the measurements at once. By a probabilistic perspective, it requires the computation of a new full posterior distribution for each instant of time t_k when the state estimate is desired [51]. When a new batch of data is available, algorithms for using the previous computed estimate as prior are available [58].

This concept can be generalised to the case when the observation batch has dimension 1 in the so-called sequential filtering approach, which will be the focus of the remainder of the chapter. The state probability distribution is updated after each new observation \mathbf{y}_k , using the previous knowledge of $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$. The update formula after a new observation is reported in Eq. (6.14). Therefore, it is directly the pre-computed conditional state distribution at t_{k-1} to be mapped at t_k , accordingly propagated with the dynamical equations, rather than the observations at different times as in the batch processor. Hence, in filtering applications, this approach results in numerical schemes, as it will be shown in detail in Sect. 6.3, able to employ efficient rules to compute the posterior $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ using the only current observation \mathbf{y}_k and the estimate at a previous time $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$, i.e. without directly taking into account the set of observations $\mathbf{y}_{1:k-1}$ and without the need to update $p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1})$ with \mathbf{y}_k . Therefore, the inversion problem dimension depends only on the number of new observations. This characteristic dramatically alleviates the computational burden associated to the computation of the posterior distribution, resulting in an efficient approach for dynamic estimation problems.

6.1.5 Optimal Estimate

The posterior conditional distribution is the solution of the filtering problem combining the a priori dynamical knowledge with the obtained measurements. According to the selected approach, this posterior could be a joint distribution, as in Eq. (6.4), or the marginal probability, in Eq. (6.5). In the general nonlinear case, this solution is infinite-dimensional.

Generally, this complete solution is not obtainable; hence a finite-dimensional approximation is sought. Furthermore, for many practical applications, a single best estimate, approximating the true state, is required out of the posterior distribution.

Intuitive choices for this statistical estimator could be the posterior's expectation, mode, median or any other statistical quantity that is faithfully representative of the true state in the statistical sense. To formalise this decision process, a *loss function* is defined as a real-value function $L(\tilde{\mathbf{x}}_k)$ quantifying a penalty (or gain) of choosing an estimate $\hat{\mathbf{x}}_k$ rather than another when approximating the true state $\bar{\mathbf{x}}_k$. Ideally, $\tilde{\mathbf{x}}_k$ is the deviation from the true state $\tilde{\mathbf{x}}_k = \mathbf{x}_k - \bar{\mathbf{x}}_k$, which however is unknown. Hence, it will be used to denote deviations from the estimate:

$$\tilde{\mathbf{x}}_k \triangleq \mathbf{x}_k - \hat{\mathbf{x}}_k . \quad (6.25)$$

Jazwinski [29] requires the loss function to satisfy the following properties:

$$\begin{aligned} L(0) &= 0 \\ \rho(\tilde{\mathbf{x}}_k^2) \geq \rho(\tilde{\mathbf{x}}_k^1) \geq 0 &\Rightarrow L(\tilde{\mathbf{x}}_k^2) \geq L(\tilde{\mathbf{x}}_k^1) \geq 0 \\ \rho &\text{ non-negative convex .} \end{aligned} \quad (6.26)$$

An intuitive choice for ρ is to be a distance measure from the zero-error origin. Given a specific loss function L , the optimal statistical decision can be formulated as an optimisation process with the goal to find the *optimal* estimate $\hat{\mathbf{x}}_k^*$, minimising the expectation of the loss function. It is worth remarking that minimising the expectation is not the only possible choice, it is just a natural and intuitive choice, likewise the most used historically. Since the true value of \mathbf{x}_k is not known, the expectation is formulated with respect to the posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ [51]:

$$\hat{\mathbf{x}}_k^* = \min_{\hat{\mathbf{x}}_k} E\{L(\tilde{\mathbf{x}}_k)|\mathbf{y}_{1:k}\} = \min_{\hat{\mathbf{x}}_k} \int L(\tilde{\mathbf{x}}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k . \quad (6.27)$$

This minimisation is equivalent to minimise the expectation of $L(\tilde{\mathbf{x}}_k)$ [29].

With this framework set, the choice of the loss function is the only factor to discriminate a specific statistical quantity. Cox [16] introduced the linear loss function:

$$L(\tilde{\mathbf{x}}_k) = \sum_i c_i |\tilde{\mathbf{x}}_k| . \quad (6.28)$$

Plugging this loss function in Eq. (6.27), it can be shown that the i -component of the optimal estimate $\hat{\mathbf{x}}_k^*$ is the median of the marginal distribution for i -component of \mathbf{x}_k conditional to the observations $\mathbf{y}_{1:k}$.

One of the most used estimator is the quadratic loss function:

$$L(\tilde{\mathbf{x}}_k) = \tilde{\mathbf{x}}_k^T W \tilde{\mathbf{x}}_k . \quad (6.29)$$

The optimal estimate of this estimator with W being the identity matrix is the conditional expectation $\hat{\mathbf{x}}_k^* = E\{\mathbf{x}_k|\mathbf{y}_{1:k}\}$ of the posterior distribution. In linear filtering, under Gaussian assumptions, this estimate coincides with the conditional mode and median. This state estimate is also called *minimum variance* estimate, because it minimises the variance for any conditional probability, or *minimum mean squared error*, as it can be derived by minimising the least square errors between computed and received observations [58].

Still Cox introduced a loss function that weights deviations larger than a set threshold equally in order to avoid few very dispersed samples to spoil the estimate, which in this framework tends to the mode of the conditional distribution.

$$L(\tilde{\mathbf{x}}_k) = \begin{cases} \|\tilde{\mathbf{x}}_k\|^2 & \text{if } \|\tilde{\mathbf{x}}_k\|^2 \leq c \\ a & \text{if } \|\tilde{\mathbf{x}}_k\|^2 > c . \end{cases} \quad (6.30)$$

The mode of the conditional distribution is exactly the state best estimate when the Dirac's delta $\delta(\cdot)$ is used in the loss function [37]:

$$L(\tilde{\mathbf{x}}_k) = 1 - \delta(\tilde{\mathbf{x}}_k) = \begin{cases} 0 & \text{if } \mathbf{x}_k = \hat{\mathbf{x}}_k \\ 1 & \text{if } \mathbf{x}_k \neq \hat{\mathbf{x}}_k . \end{cases} \quad (6.31)$$

This loss function choice yields the *maximum a posteriori* (MAP) estimator as the minimising point is the peak of the posterior distribution. This can be seen as a particular case of the function in Eq. (6.30), when no loss is associated to the correct point and equal loss to any deviation from it.

For the scalar case, Kalman [36] introduced a quartic loss function $L(\tilde{x}_k) = a\tilde{x}_k^4$ and an exponential one $L(\tilde{x}_k) = a[1 - \exp(-\tilde{x}_k^2)]$.

If $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is symmetric about its conditional expectation and unimodal, the optimal state estimation is the conditional expectation $\hat{\mathbf{x}}_k^* = E\{\mathbf{x}_k|\mathbf{y}_{1:k}\}$ for every loss function L satisfying the properties in Eq. (6.26) [29]. Therefore the conditional expectation is the chosen estimate for a large variety of filtering problems.

During this theoretical development, the process of deriving an optimal estimate relied on the assumption that a full posterior distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ would be available. However, in the general nonlinear filtering case, it is often impracticable to derive the full posterior distribution. To deal with this issue, practical methods to efficiently compute the first moments of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ have been developed, and they will be presented in later sections. Hence, if the state estimate is chosen as the conditional expectation, no further calculation is needed as $\hat{\mathbf{x}}_k^*$ coincides with the first moment of the posterior distribution. On the other hand, we cannot just look for the conditional mean. First, it often depends on higher-order moments. Second, higher-order moments are an indication of how the probability is dispersed around the mean value, therefore providing a measure of how accurate the estimate represents the distribution. In the words of Jazwinski: 'It can be argued that knowledge of the second-order moment is just as important as knowing the estimate itself. An estimate is meaningless unless one knows how good it is.'

6.2 Probability Distribution Propagation

This section will present practical tools for the computation of the evolution of density functions, presented in the previous theoretical section, when propagated through arbitrary transformations. In particular, the main difficulty stems from the general nonlinearity of the dynamical equations $\mathbf{f}(t, \mathbf{x})$ and observation relationships $\mathbf{g}(t, \mathbf{x})$. Indeed, in the linear case, the probability propagation has a closed-form solution, as it will be shown in Sect. 6.2.1. On the other hand, there is no analytical solution for the general nonlinear case, and approximations shall be introduced to compute a solution, as shown in Sect. 6.2.2.

6.2.1 Linear Transformation

In the linear time-varying case, the equations describing the evolution of the distribution moments take a simplified form. Indeed, if the dynamical equations can be written as

$$\dot{\mathbf{x}} = F(t)\mathbf{x} + G(t)\mathbf{w}, \quad (6.32)$$

with \mathbf{w} the white Gaussian noise, the Eqs. (6.22) and (6.23) simplify without approximations to:

$$\frac{d\hat{\mathbf{x}}}{dt} = F(t)\hat{\mathbf{x}} \quad (6.33)$$

$$\frac{dP_x}{dt} = F(t)P_x + P_x F^T(t) + G(t)QG(t)^T, \quad (6.34)$$

where again Q is the covariance of the dynamical noise w . This form describes the exact evolution of the first two moments of the density function in a linear system, and it is the basis of the linear filtering (see Sect. 6.3.1). It is worth noting that these are ordinary differential equations and therefore are relatively easy to integrate numerically. Specifically, the first equation implies that the mean of the propagated distribution is the propagated mean of the initial distribution. The second equation describes in compact matrix notation how the covariance matrix evolves as result of the deterministic term and the process noise.

In linear filtering, also the observation model is linear:

$$\mathbf{y} = H(t)\mathbf{x} + \boldsymbol{\epsilon}. \quad (6.35)$$

Hence, the probability $p(\mathbf{y}_k|\mathbf{x}_k)$ of the measurements conditional to the state for Gaussian distributions (see also the reasoning in Sect. 6.1.3 for general density functions) is given by:

$$p(\mathbf{y}_k | \mathbf{x}_k) \sim \mathcal{N}_{\mathbf{y}_k}(H(t_k)\mathbf{x}_k, R_k), \quad (6.36)$$

where R_k is the covariance of the measurement noise.

If the prior density function is Gaussian,

$$\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, P_0), \quad (6.37)$$

all the densities in the update step via Bayes's rule (see Eq. (6.14)) are Gaussian as well. Therefore, the first two moments' evolutions between observations, and at an observation update, completely characterise the distributions.

6.2.2 Nonlinear Transformation

In the majority of applications, the filtering model involves nonlinear dynamical equations and observation relationships. The major drawback for filtering is that when a Gaussian density is plugged in a nonlinear relationship, it loses its Gaussianity. In general, for \mathbf{x} a random variable with $p_x(\mathbf{x})$, the random variable $\mathbf{z} = \mathbf{g}(\mathbf{x})$ has density function [29]:

$$p_z(\mathbf{z}) = p_x(\mathbf{g}^{-1}(\mathbf{z})) \left| \det \left(\frac{\partial \mathbf{g}^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|, \quad (6.38)$$

for invertible \mathbf{g} . Generally, it is not possible to solve directly for this non-Gaussian distribution. Often, numerical filter techniques rely on the Gaussian approximation of this density to simplify the filtering computation. As a Gaussian distribution is completely defined by its mean and covariance, a variety of methods exist to compute directly these first two moments of the derived distribution. In this section, the relation $\mathbf{g}(\mathbf{x})$ indicates an arbitrary function, which can represent both the observation model and the discrete dynamical transition step. In the latter case, this can be a state transition operator or the result of a numerical integration scheme.

This section will first present methods based on Taylor's expansion of the nonlinear transformation, in Sect. 6.2.2.1. Then, methods based on sample propagation will be shown, specifically with deterministic sampling in Sect. 6.2.2.2 and with random sampling in Sect. 6.2.2.3.

6.2.2.1 Taylor Expansion

The nonlinear transformation \mathbf{g} can be expanded in Taylor's series about the expected value $\hat{\mathbf{x}} = E\{\mathbf{x}\}$:

$$\mathbf{z} = \mathbf{g}(\mathbf{x}) = \mathbf{g}(\hat{\mathbf{x}}) + \nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}}) + \frac{1}{2} \sum_i (\mathbf{x} - \hat{\mathbf{x}})^T H_{\mathbf{xx}}^{(i)}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}})\mathbf{e}_i + \mathcal{O}\left((\mathbf{x} - \hat{\mathbf{x}})^3\right), \quad (6.39)$$

where $\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}$ is the Jacobian matrix of \mathbf{g} , while $H_{\mathbf{xx}}^{(i)}|_{\hat{\mathbf{x}}}$ is the Hessian matrix of i -component of \mathbf{g} , both evaluated at $\hat{\mathbf{x}}$, and \mathbf{e}_i is a unit vector pointing along the i -coordinate axis.

Truncating at the first order, the random variable \mathbf{z} has now a simple expression for its mean [51]:

$$\begin{aligned} \hat{\mathbf{z}} &\approx E\{\mathbf{g}(\hat{\mathbf{x}}) + \nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}})\} \\ &= E\{\mathbf{g}(\hat{\mathbf{x}})\} + E\{\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}})\} \\ &= \mathbf{g}(\hat{\mathbf{x}}) + \nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}E\{(\mathbf{x} - \hat{\mathbf{x}})\} \\ &= \mathbf{g}(\hat{\mathbf{x}}). \end{aligned} \quad (6.40)$$

This result shows that, at first-order approximation, the expected value of the transformed distribution is the expected value of the input distribution propagated through the nonlinear equation. Using this approximation, the covariance matrix becomes:

$$\begin{aligned} P_z &= E\left\{(\mathbf{g}(\mathbf{x}) - \hat{\mathbf{z}})(\mathbf{g}(\mathbf{x}) - \hat{\mathbf{z}})^T\right\} \\ &\approx E\left\{(\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}}))(\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}}))^T\right\} \\ &= \nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}E\left\{(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right\}\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}^T \\ &= \nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}P_x\nabla_{\mathbf{x}}\mathbf{g}|_{\hat{\mathbf{x}}}^T. \end{aligned} \quad (6.41)$$

This first-order approximation is the basis for the classical version of the *extended Kalman filter* (see Sect. 6.3.2).

However, when the model is highly nonlinear or the deviations $(\mathbf{x} - \hat{\mathbf{x}})$ are significant, this approximation can become too inaccurate for the application requirements. To better capture the nonlinear function's behaviour, second-order terms in the Taylor expansion can be retained. Hence, the expected value becomes:

$$\begin{aligned} \hat{\mathbf{z}} &\approx \mathbf{g}(\hat{\mathbf{x}}) + E\left\{\frac{1}{2} \sum_i (\mathbf{x} - \hat{\mathbf{x}})^T H_{\mathbf{xx}}^{(i)}|_{\hat{\mathbf{x}}}(\mathbf{x} - \hat{\mathbf{x}})\mathbf{e}_i\right\} \\ &= \mathbf{g}(\hat{\mathbf{x}}) + \frac{1}{2} \sum_i \text{tr}\left(H_{\mathbf{xx}}^{(i)}|_{\hat{\mathbf{x}}}P_x\right)\mathbf{e}_i. \end{aligned} \quad (6.42)$$

The trace operator appears as the quadratic form is a scalar quantity, and its cyclic property is exploited to obtain the final form. For a detailed derivation, see Mathai and Provost [41]. In this case, the expected value of the transformed density depends on the second-order moment of the initial distribution through second-order derivatives of the nonlinear function \mathbf{g} , under the obvious assumption that \mathbf{g} is twice differentiable. The covariance matrix of \mathbf{z} is [27]:

$$P_z \approx \nabla_{\mathbf{x}} \mathbf{g} \Big|_{\hat{\mathbf{x}}} P_x \nabla_{\mathbf{x}} \mathbf{g} \Big|_{\hat{\mathbf{x}}}^T + \frac{1}{2} \sum_{i,j} \text{tr} \left(P_x H_{\mathbf{xx}}^{(i)} \Big|_{\hat{\mathbf{x}}} P_x H_{\mathbf{xx}}^{(j)} \Big|_{\hat{\mathbf{x}}} \right) \mathbf{e}_i \mathbf{e}_j^T. \quad (6.43)$$

These terms could be used as compensation for the neglected quadratic effects in the classical *extended Kalman filter*.

6.2.2.2 Unscented Transform

The Taylor expansion, and the consequent Extended Kalman Filter, involves the linearisation of the dynamics. This can cause poor performance or filter divergence when the dynamics is highly nonlinear or the initial conditions are known with low accuracy. Furthermore, the Taylor expansion requires the explicit derivation of derivatives, which is not always possible. Even when the functional dependencies of \mathbf{g} are explicitly known, this requirement makes the numerical system error-prone.

To solve these issues, Julier and Uhlmann developed a new technique to approximate nonlinear transformations of the probability distribution functions. They started from the intuition that it should be easier to approximate a normal distribution than an arbitrary nonlinear function [31]. Indeed, instead of expanding the transformation $\mathbf{g}(\mathbf{x})$, the considered alternative is to approximate the output distribution $p(\mathbf{z})$ directly based on a set of response samples.

This recent technique, named *unscented transformation*, fits a discrete distribution of N_σ sigma points \mathbf{x}_i to the initial density $p(\mathbf{x})$. A weight w_i , positive or negative, is associated to each sigma point with the condition $\sum_i w_i = 1$ to have an unbiased estimate. Once this set of deterministic samples has been selected, they are propagated through the nonlinear function $\mathbf{z}_i = \mathbf{g}(\mathbf{x}_i)$, and from them the posterior density moments are reconstructed [34]:

$$\hat{\mathbf{z}} \approx \sum_{i=1}^{N_\sigma} w_i \mathbf{z}_i \quad (6.44)$$

$$P_z \approx \sum_{i=1}^{N_\sigma} w_i (\mathbf{z}_i - \hat{\mathbf{z}})(\mathbf{z}_i - \hat{\mathbf{z}})^T. \quad (6.45)$$

As computing the moments of the resulting distribution is rather straightforward, the key passage turns out to be the selection process of the sigma points and the

associated weights. In the general approach, this selection process can be seen as a constrained optimisation problem where the number of samples, the associated weights and their position are the free variables [10]. The constraints are imposed to meet the requirement that the discrete distribution, generated by the selected weights and sigma points, reproduces important statistical characteristics of $p(\mathbf{x})$. As generally the number of free parameters can be higher than the number of constraints, the remaining parameters can be used to minimise a penalty function, e.g. higher-order moments deviation.

This general approach resulted in the birth of numerous variants of unscented filters. The computational cost of the unscented transformation is proportional to the number of sigma points employed, so there is a propensity to choose schemes with only few degrees of freedom. Among them, the simplex unscented approach requires a minimum number of $N_x + 1$ samples to match the mean and covariance of a N_x dimensional normally distributed random vector \mathbf{x} [30, 33]. On the other hand, additional sigma points can be introduced to reproduce higher-order moments of a Gaussian distribution, e.g. $2N_x^2 + 1$ points are required to match up to the fourth-order moment (kurtosis) with a penalty function minimising the sixth-order moment [32]. The most used variant relies on the use of $2N_x + 1$ sigma points [31]. This unscented transformation is able to approximate a Gaussian distribution up to the third-order, while errors appear as a result of fourth-order cross-kurtoses terms. In the derivation by Wan and Van Der Merwe, the points and weights are selected symmetrically around the mean value as [63]:

$$\begin{aligned} \mathbf{x}_0 &= \hat{\mathbf{x}} \\ \mathbf{x}_i &= \hat{\mathbf{x}} + \sqrt{(N_x + \lambda) P_x^{(i)}} \quad \text{for } i = 1, \dots, N_x \end{aligned} \quad (6.46)$$

$$\begin{aligned} \mathbf{x}_i &= \hat{\mathbf{x}} - \sqrt{(N_x + \lambda) P_x^{(i-N_x)}} \quad \text{for } i = N_x + 1, \dots, 2N_x \\ w_0 &= \lambda / (N_x + \lambda) \\ w_i &= 1 / [2(N_x + \lambda)] \quad \text{for } i = 1, \dots, 2N_x, \end{aligned} \quad (6.47)$$

where λ is a scaling parameter and $P_x^{(i)}$ is the i -th column of the covariance matrix of \mathbf{x} . The free scaling parameter can be chosen to minimise the deviation of the kurtosis. This parameter is often rewritten as $\lambda = \alpha^2(N_x + k) - N_x$ to better control the covariance positive definiteness [10], where α tunes the sigma point spread about the mean, while k can be used either to incorporate knowledge about higher moments of the starting distribution or to minimise their deviation. This reparameterisation causes a change in the weight for the central sample, which now is $w_0^{(m)} = \lambda / (N_x + \lambda)$ when computing the mean, therefore used in Eq. (6.44), while $w_0^{(c)} = \lambda / (N_x + \lambda) + (1 - \alpha^2 + \beta)$, used in Eq. (6.45). The parameter β can be used to incorporate a priori knowledge on the distribution of the initial variable, e.g. $\beta = 2$ for Gaussian \mathbf{x} [60, 63]. In the case of Gaussian initial distribution, the

sigma points of this variant faithfully capture the mean and covariance, while the transformed probability distribution reconstructed from the propagated samples has exact mean for polynomial $\mathbf{g}(\mathbf{x})$ up to degree three and exact covariance for $\mathbf{g}(\mathbf{x})$ linear [51].

The main advantage of the unscented transformation is that it does not require differentiability, or derivative information, of the nonlinear mapping \mathbf{g} but only the propagation of a limited number of deterministic samples. It is worth stressing that the Gaussianity approximation on the prior and updated distributions is not a required assumption of the unscented transformation. Nonetheless, the majority of practical filters employ this transformation with Gaussian distributions only, mainly because of the simplification in the Bayes' step.

6.2.2.3 Monte Carlo Methods

Another large family of techniques to approximate the posterior density is *Monte Carlo methods*. Unlike unscented transformation methods, the set of samples is generated randomly according to a given distribution. Therefore, this method does not require any linearity or Gaussian assumption on the model. Furthermore, unlike deterministic methods, the number of samples required for the mean to converge is theoretically independent of the problem's dimensionality [38]. With the propagated samples, the moments are estimated as [27]:

$$\hat{\mathbf{z}} \approx \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i = \frac{1}{N} \sum_{i=1}^N \mathbf{g}(\mathbf{x}_i) \quad (6.48)$$

$$P_z \approx \frac{1}{N-1} \sum_{i=1}^N (\mathbf{z}_i - \hat{\mathbf{z}})(\mathbf{z}_i - \hat{\mathbf{z}})^T. \quad (6.49)$$

In this conventional Monte Carlo, it is evident to infer how crucial it is to properly select the random samples \mathbf{x}_i in accordance to the original probability distribution $p(\mathbf{x})$. This is numerically straightforward when $p(\mathbf{x})$ is Gaussian or belongs to any simple distribution family. However, in Bayesian filtering, it is usually numerically demanding to sample directly from the required density because of its complex functional form (see Eq. (6.19)).

Markov chain Monte Carlo techniques are a class of efficient methods to generate the random samples from a distribution $p(\mathbf{x})$. The basic concept is to replace the target density sampling by a Markov chain which has $p(\mathbf{x})$ as equilibrium distribution, and sample a realisation of this instead. In the literature, there is an abundance of algorithms, mainly differing by the transition Kernel used for the Markov chain. The first example is the notable Metropolis algorithm [44]. Extensive references are provided by Gilks et al. [23] or Brooks et al. [8].

Another class of methods is *importance sampling*, which draws from an approximated density $\pi(\mathbf{x})$, simpler to sample, instead of the original $p(\mathbf{x})$. Then, the

moments can be computed as in Eqs. (6.48) and (6.49) by weighting each sample with a measure of the deviation between the original and sampled distribution. The requirement on the *importance* density is that its support should be greater or equal to the one of $p(\mathbf{x})$ [38]. To compute the weight, we can decompose the expectation formula as [51]:

$$\begin{aligned} E\{\mathbf{z}\} &= E_p\{\mathbf{g}(\mathbf{x})\} = \int \mathbf{g}(\mathbf{x})p(\mathbf{x})d\mathbf{x} \\ &= \int \mathbf{g}(\mathbf{x})\frac{p(\mathbf{x})}{\pi(\mathbf{x})}\pi(\mathbf{x})d\mathbf{x} = E_\pi\left\{\mathbf{g}(\mathbf{x})\frac{p(\mathbf{x})}{\pi(\mathbf{x})}\right\}. \end{aligned} \quad (6.50)$$

Therefore, as the samples \mathbf{x}_i are generated using $\pi(\mathbf{x})$, their function evaluation should be weighted by $p(\mathbf{x})/\pi(\mathbf{x})$. The approximation of the expected value of \mathbf{z} now becomes:

$$\hat{\mathbf{z}} \approx \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i = \frac{1}{N} \sum_{i=1}^N \frac{p(\mathbf{x}_i)}{\pi(\mathbf{x}_i)} \mathbf{g}(\mathbf{x}_i) = \sum_{i=1}^N w_i \mathbf{g}(\mathbf{x}_i). \quad (6.51)$$

Hence, the weight of the i th sample is:

$$w_i = \frac{1}{N} \frac{p(\mathbf{x}_i)}{\pi(\mathbf{x}_i)}. \quad (6.52)$$

Intuitively, the weights correct the bias associated to the samples selected from a nonideal distribution. Clearly, the closer the *importance* distribution is to the original one, the smaller the required bias correction is, i.e. $w_i \approx 1/N$. The same weights computed for the expected value can be used for the covariance or higher-order moment approximation. Liu [38] suggests to use the normalised weights:

$$w_i^* = \frac{w_i}{\sum_i w_i}. \quad (6.53)$$

The resulting estimate, although biased, often results in a smaller mean squared error. The same weight choice is found in Sarkka [51] when deriving the importance sampling form for conditional probabilities (see Sect. 6.3.5).

Indeed, this framework applies also for the density functions in the sequential filtering algorithms, where $p(\mathbf{x})$ and $\pi(\mathbf{x})$ are substituted by conditional distributions. In an attempt to connect the generic notation above to the filtering problem of interest, the vector \mathbf{x} can be decomposed as $\mathbf{x} = \mathbf{x}_{0:k} = [\mathbf{x}_0, \dots, \mathbf{x}_k]$. Hence, looking at Eq. (6.50) with $\mathbf{x}_{0:k}$ and substituting a density conditional on measurements $\mathbf{y}_{1:k}$, we obtain the importance sampling approximation to the expectation operator of the sequential filtering posterior distribution. However, the basic importance sampling approach is not well-suited for sequential filtering approaches. Indeed, when computing $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ by the importance distribution $\pi(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, it would

be advantageous to exploit the previous density $\pi(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$. The same holds for the marginal distribution with respect to the current state. For this reason, sequential filters employ the *sequential importance sampling* variant [20]. Given the decomposition of $\mathbf{x} = \mathbf{x}_{0:k}$, the joint target density can be written as the product of conditional densities as in Eq. (6.10). The importance distribution can be written in a similar form:

$$\pi(\mathbf{x}_{0:k}) = \pi(\mathbf{x}_0) \prod_{j=1}^k \pi(\mathbf{x}_j|\mathbf{x}_{0:j-1}). \quad (6.54)$$

Hence, the formula for the weight for a specific sample is given by:

$$w_k = \frac{p(\mathbf{x}_0) \prod_{j=1}^k p(\mathbf{x}_j|\mathbf{x}_{0:j-1})}{\pi(\mathbf{x}_0) \prod_{j=1}^k \pi(\mathbf{x}_j|\mathbf{x}_{0:j-1})}, \quad (6.55)$$

where the multiplicative constant has been ignored for now. It is straightforward to see how this approach suits the sequential estimation case. Indeed, by defining $w_0 = p(\mathbf{x}_0)/\pi(\mathbf{x}_0)$, the recursive formula immediately follows:

$$w_k = w_{k-1} \frac{p(\mathbf{x}_k|\mathbf{x}_{0:k-1})}{\pi(\mathbf{x}_k|\mathbf{x}_{0:k-1})}. \quad (6.56)$$

An equivalent approach can be derived for probabilities conditional on measurements, as needed by filtering approaches [10], which will be presented in Sect. 6.3.5.

One major and quite frequent issue encountered in sequential importance sampling is the degeneracy of the weights, i.e. when most of the particles have an irrelevant weight. This effect is caused by the increase of the weight variance with iterations [18]. Resampling routines add to the sequential importance sampling, a step in which a subset of particles is substituted by new ones drawn from the current weighted approximation of the density function. The resampling approach for optimal filtering with a sequential importance algorithm has been introduced by Gordon with the *Bootstrap filter* [24]. Plenty of variants exist, both on the importance distribution selection and on the resampling techniques, in the broad family of *Sequential Monte Carlo* methods. This class is well-suited for sequential filtering problems in which the density functions rapidly vary in time [18]. A detailed discussion is beyond the scope of this chapter, yet the interesting reader can consult the existing comprehensive literature, e.g. Liu [38] or Doucet [20].

6.3 Filtering Algorithms

If the dynamical equations and observation model are time-varying linear and the prior density distribution of \mathbf{x}_0 is Gaussian, all the involved probabilities between

observations and after an observation update will retain a normal distribution (see Sect. 6.2.1). This precious characteristic and the simple combination rules of Gaussian distributions result in a closed-form exact solution of the filtering equations called *Kalman filter* (KF), introduced in Sect. 6.3.1. However, in the majority of real-world applications, the involved transformations are nonlinear, and a closed-form solution does not exist in the general case. Nonetheless, the Gaussian approximation of the conditional density proves sufficient for a wide range of practical applications. To approximate the Gaussian evolution through a nonlinear transformation, the techniques introduced in the previous section shall be used. Specifically, the *extended Kalman filter* (EKF), presented in Sect. 6.3.2, expands the nonlinear function in Taylor series and retains only the first terms, whereas the *unscented Kalman filter* (UKF), described in Sect. 6.3.3, approximates the posterior distribution using the unscented transformation deterministic approach.

The basic idea of approximating the probability density function as being normally distributed has been embedded in the general framework of *Gaussian filtering*. This denomination encloses a family of algorithms employing moment matching approximations, and usually explicit cubature rules for computing the integrals required by the expectation operator. As it turns out, the general Gaussian filtering framework can be seen as a generalisation of the Kalman filter and some of its extensions presented in this chapter. This framework will be shortly outlined in Sect. 6.3.4.

When the assumption of normal densities is too restrictive or not representative, other techniques should be employed to approximate the underlying real distribution in a finite-dimensional basis. Among the several existing methods, the particle filter employs sampling methods, hence resulting in a discrete distribution. As a sampling-based method, the particle filter is highly flexible and capable of approximating posterior distribution of any nature, when the number of samples is selected appropriately. This filter will be introduced in Sect. 6.3.5.

6.3.1 *Kalman Filter*

In the linear case, the filtering model is described by the linear equations of motion and observation model. In general, linear systems are rather easy to characterise and often allow closed-form solutions. On the other hand, they can model only simplified problems, as real-world systems generally involve complex nonlinearities. Nonetheless, the closed-form solution of an associated linear system can be used to construct an approximation of the original one, or in general can provide useful insight in some of its properties.

The Kalman filter is the closed-form algorithm for the evolution of the conditional probability density in a sequential linear filtering problem. The model for the linear filtering problem definition (see Sect. 6.1.1) is formulated as:

$$\begin{aligned}
\dot{\mathbf{x}} &= F(t)\mathbf{x} + G(t)\mathbf{w} \\
\mathbf{y} &= H(t)\mathbf{x} + \boldsymbol{\epsilon} \\
\mathbf{x}_0 &\sim \mathcal{N}_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, P_0) .
\end{aligned} \tag{6.57}$$

At time t_0 , before any observation, the conditional distribution coincides with the prior distribution

$$p(\mathbf{x}_0|\mathbf{y}_0) = p(\mathbf{x}_0) = \mathcal{N}_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, P_0) , \tag{6.58}$$

where \mathbf{y}_0 has been introduced for notation's consistency, but it is a fictitious quantity. For generality, the derivation process will be carried out starting from a generic time t_{k-1} and distribution $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ after an observation has been processed, which could be also the initial time for $k = 1$ thanks to the fictitious observation introduced.

The density distribution can be propagated to represent the state probability distribution at a given time of interest. In the general case, the Kolmogorov partial differential equation should be used (see Eq. (6.21)). However, in Sect. 6.2.1, it has been shown how the first two moments evolve according to simple ordinary differential equations in the linear case. As the initial condition is given by a normally distributed density, the first two moments fully capture the statistics of the conditional density. Therefore, when propagating at the time of the next observation, the conditional probability is [29]:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-) , \tag{6.59}$$

where the superscript $\{\cdot\}^-$ has been introduced to describe a quantity at an infinitesimal time before t_k , i.e. just before the observation update. Similarly, the superscript $\{\cdot\}^+$ will be used to identify a quantity at an infinitesimal time after t_k , i.e. immediately after the update with a new measurement. The moments $\hat{\mathbf{x}}_k^-$ and P_k^- can be obtained by numerical propagation of the ordinary differential equations (6.33)–(6.34), respectively, with initial conditions $\hat{\mathbf{x}}_{k-1}$ and P_{k-1} .

When an observation is available, this new knowledge is combined with the dynamically propagated distribution to obtain a better estimate of the state. The Kalman filter updates the conditional state distribution through Bayes' rule, in the sequential filtering form of Eq. (6.14). The second probability in the numerator is computed as in Eq. (6.59). The density of the observation, conditional on the state immediately before the observation, is given by Eq. (6.36), written here as $p(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}_{\mathbf{y}_k}(H(t_k)\mathbf{x}_k, R_k)$. Lastly, the denominator could be decomposed as in Eq. (6.18). However, instead of computing the quantity $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ by integration, we can first exploit a well-known property for computing the joint distribution of two Gaussian random variables with conditional dependencies [51]:

$$\begin{aligned}
p(\mathbf{y}_k, \mathbf{x}_k | \mathbf{y}_{1:k-1}) &= p(\mathbf{y}_k | \mathbf{x}_k, \mathbf{y}_{1:k-1}) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \\
&= p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) \\
&= \mathcal{N}_{\mathbf{y}_k}(H(t_k)\mathbf{x}_k, R_k) \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-) \\
&= \mathcal{N}_{\mathbf{x}_k, \mathbf{y}_k} \left(\begin{pmatrix} \hat{\mathbf{x}}_k^- \\ H(t_k)\hat{\mathbf{x}}_k^- \end{pmatrix}, \begin{pmatrix} P_k^- & P_k^- H_k^T \\ H_k P_k^- & H_k P_k^- H_k^T + R_k \end{pmatrix} \right). \tag{6.60}
\end{aligned}$$

Then, as their joint distribution is Gaussian, the marginal distribution of \mathbf{y}_k is simply computed as:

$$p(\mathbf{y}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{y}_k} \left(H(t_k)\hat{\mathbf{x}}_k^-, H_k P_k^- H_k^T + R_k \right). \tag{6.61}$$

This simple marginalisation rule stems from the definition of multivariate normal distributions and the linear algebra operators involved. With all the densities in Eq. (6.14) derived, the resulting distribution via Bayes' inference can be derived by multiplication and division rules between normal distributions, leading to the following result [29]:

$$p(\mathbf{x}_k | \mathbf{y}_k) = \frac{\mathcal{N}_{\mathbf{y}_k}(H(t_k)\mathbf{x}_k, R_k) \cdot \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)}{\mathcal{N}_{\mathbf{y}_k}(H(t_k)\hat{\mathbf{x}}_k^-, H_k P_k^- H_k^T + R_k)} = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+), \tag{6.62}$$

where

$$\hat{\mathbf{x}}_k^+ = \left(H^T R^{-1} H + P_k^{-1} \right)^{-1} \left(H^T R_k^{-1} \bar{\mathbf{y}}_k + P_k^{-1} \hat{\mathbf{x}}_k^- \right) \tag{6.63}$$

$$P_k^+ = \left(H^T R^{-1} H + P_k^{-1} \right)^{-1}. \tag{6.64}$$

These equations express the update step in the linear sequential filtering algorithm after the observation value $\bar{\mathbf{y}}_k$ is received. However, this form requires the inversion of a square matrix of dimension equal to the number of state variables. By matrix operations, the update step can be reduced to:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\bar{\mathbf{y}}_k - H \hat{\mathbf{x}}_k^-) \tag{6.65}$$

$$P_k^+ = P_k^- - K_k H P_k^-, \tag{6.66}$$

where K_k is the well-known *Kalman gain* defined as:

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1}. \tag{6.67}$$

This algorithmic variant requires the inversion of a square matrix of a dimension equal to the number of new observations, which in practical applications is smaller

than the state dimension. Furthermore, if the observations are uncorrelated, i.e. if R_k is diagonal, the observations can be processed one by one, therefore requiring only a scalar division, eventually leading to the same estimate obtained by processing the batch $\bar{\mathbf{y}}_k$ at once [29]. Therefore, for observations the algorithm can now be schematised as in Algorithm 1. The Kalman Filter is one of the very few closed-

Algorithm 1 Kalman Filter

- Given the filtering model in Eq. (6.57)
- 1: Initialise $t_{k-1} = t_0$, $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_0$, $P_{k-1}^+ = P_0$, $t_k = t_1$
 - 2: **for** Observation times **do**
 - Prediction step:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)$
 - 3: Propagate mean with $\hat{\dot{\mathbf{x}}} = F\hat{\mathbf{x}}$
 $\hat{\mathbf{x}}_{k-1}^+ \rightarrow \hat{\mathbf{x}}_k^-$
 - 4: Propagate covariance with $\dot{P} = FP_x + P_x F^T + GQG^T$
 $P_{k-1}^+ \rightarrow P_k^-$
 - Update step:** after observation $\bar{\mathbf{y}}_k$ compute $p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+)$
 - 5: Compute Kalman gain
 $K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1}$
 - 6: Update mean with observation information
 $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\bar{\mathbf{y}}_k - H\hat{\mathbf{x}}_k^-)$
 - 7: Update covariance with observation covariance
 $P_k^+ = P_k^- - K_k H P_k^-$
 - 8: Update quantities for loop iteration
 $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_k^+$, $P_{k-1}^+ = P_k^+$, $k = k + 1$
 - 9: **end for**
-

form solutions of the general sequential filtering equations. Although it relies on restrictive assumptions, historically it had a crucial role in the development of approximated numerical techniques which are employed in real-world applications. This version of the Kalman filter is likely the most simple for gaining insight in its prediction-update sequential scheme. However, for actual numerical implementation, Algorithm 1 is not the most robust alternative. Indeed, numerical errors could cause the covariance matrix to lose its symmetry and positive definiteness properties [58]. Rather than Eq. (6.66), alternative updates for the covariance matrix are proposed to have a better numerical stability [4, 7, 9, 58]. For a specialised and instructive overview on the practical algorithm variants for the Kalman filter, the reader is reminded to specific references [25].

One formulation that is worth to discuss is the one employing the state transition matrix to propagate the state. Indeed, as anticipated in the previous section, the dynamics of a linear system can be formulated by integral equations as [29]:

$$\mathbf{x}_k = \Phi(t_k, t_{k-1})\mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} \Phi(t_k, \tau)G(\tau)d\mathbf{w}. \quad (6.68)$$

where the integral term has zero-mean. The state transition matrix evolution is given by:

$$\dot{\Phi}(t, t_{k-1}) = F(t)\Phi(t, t_{k-1}), \text{ with } \Phi(t_{k-1}, t_{k-1}) = I. \quad (6.69)$$

In this formulation, Eqs. (6.33) and (6.34) can be reformulated as [58]:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \Phi(t_k, t_{k-1})\hat{\mathbf{x}}_{k-1} \\ P_x(t_k) &= \Phi(t_k, t_{k-1})P_x(t_{k-1})\Phi^T(t_k, t_{k-1}) \\ &\quad + \int_{t_{k-1}}^{t_k} \Phi(t_k, \tau)G(\tau)Q(\tau)G^T(\tau)\Phi^T(t_k, \tau)d\tau. \end{aligned} \quad (6.70)$$

When \mathbf{w} is approximated to be a random sequence, i.e. piecewise constant $\mathbf{w}(t) = \mathbf{w}_k$ for $t_{k-1} \geq t \geq t_k$, with covariance Q_{k-1} , Eq. (6.68) can be written as:

$$\mathbf{x}_k = \Phi(t_k, t_{k-1})\mathbf{x}_{k-1} + \Gamma(t_k, t_{k-1})\mathbf{w}_{k-1}, \quad (6.71)$$

with $\Gamma(t_k, t_{k-1}) = \int_{t_{k-1}}^{t_k} \Phi(t_k, \tau)G(\tau)d\tau$ which can be computed by quadrature. The second term of Eq. (6.70) becomes:

$$P_x(t_k) = \Phi(t_k, t_{k-1})P_x(t_{k-1})\Phi^T(t_k, t_{k-1}) + \Gamma(t_k, t_{k-1})Q_{k-1}\Gamma^T(t_k, t_{k-1}), \quad (6.72)$$

where $\Gamma(t_k, t_{k-1})$ is called *process noise transition matrix* [58].

The corresponding algorithm for the state transition matrix approach is schematised in Algorithm 2. This procedure and its corresponding algorithm show that a linear filtering problem with continuous dynamics and discrete observations can be translated into an equivalent fully discrete linear filtering problem. This equivalence in linear filtering problems stands as an intuitive, although not formal, proof of the applicability of the techniques for nonlinear transformation approximation introduced in the previous section to the continuous-discrete filtering problem.

As a final note of the discussion on the Kalman filter, it is worth discussing the nature of the noise term in the practical applications. Indeed, since in aerospace applications the dynamics is regarded as deterministic, although not completely known or modelled, the term \mathbf{w} is not a proper Gaussian noise. Nonetheless, in such cases, it is just a useful tool for taking into account errors arising from unmodelled terms, neglected nonlinearities, numerical errors and so on [29].

In statistical derivations, this term is often left out, resulting in the covariance matrix evolution being determined exclusively by the deterministic terms (step 4 of Algorithm 1, step 5 of Algorithm 2). Then, a rightful doubt could arise whether the term \mathbf{w} is actually necessary for similar applications. Clearly, it introduces analytical difficulties in the filter derivation and numerical complexity in the algorithm. Nevertheless, it turns out that the additive term in the covariance propagation helps the filter accuracy and general performance. Indeed, without the random

Algorithm 2 Kalman Filter with state transition matrix

Given the filtering model in Eq. (6.57)

- 1: Initialise $t_{k-1} = t_0$, $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_0$, $P_{k-1}^+ = P_0$, $t_k = t_1$
 - 2: **for** Observation times **do**
 - Prediction step:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)$
 - 3: Propagate state transition matrix $\dot{\Phi}(t, t_{k-1}) = F(t)\Phi(t, t_{k-1})$
 $I \rightarrow \Phi(t_k, t_{k-1})$
 - 4: Propagate mean estimate
 $\hat{\mathbf{x}}_k^- = \Phi(t_k, t_{k-1})\hat{\mathbf{x}}_{k-1}^+$
 - 5: Propagate covariance matrix
 $P_k^- = \Phi(t_k, t_{k-1})P_{k-1}^+\Phi^T(t_k, t_{k-1}) + \Gamma(t_k, t_{k-1})Q_{k-1}\Gamma^T(t_k, t_{k-1})$
 - Update step:** after observation $\bar{\mathbf{y}}_k$ compute $p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+)$
 - 6: Compute Kalman gain
 $K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1}$
 - 7: Update mean with observation information
 $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\bar{\mathbf{y}}_k - H\hat{\mathbf{x}}_k^-)$
 - 8: Update covariance with observation covariance
 $P_k^+ = P_k^- - K_k H P_k^-$
 - 9: Update quantities for loop iteration
 $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_k^+$, $P_{k-1}^+ = P_k^+$, $k = k + 1$
 - 10: **end for**
-

term, the state covariance matrix P_k could approach zero when the number of processed observations is quite large. In such cases, the covariance trace slightly increases during propagation between observations, and it drops during the update step by the quantity $\text{tr}(K_k H P_k^-)$, i.e. depending on the accuracy of the processed observation [58]. A high number of accurate observations could therefore result in an asymptotically zero state covariance matrix, i.e. the belief that the state estimate is extremely accurate. This results directly in a small Kalman gain and therefore causes the filter state estimate to become insensitive to new observations. This will cause the filter to diverge due to neglected dynamical nonlinearities (introduced in the next section) or unmodelled terms [52]. On the other hand, if the noise term is employed, the state covariance matrix will asymptotically approach the non-zero noise covariance value. Hence, the filter state estimate will always remain sensitive to new observations. Intuitively, in practical applications, the process noise expedient is used to account for the neglected dynamics by explicitly telling the filter that its dynamical knowledge is imperfect.

6.3.2 Extended Kalman Filter

Real-world state estimation problems usually involve nonlinear dynamical and measurement models, and the Kalman filter cannot be directly applied. In the previous section, several methods for approximating nonlinear transformation were introduced. One straightforward approach is to linearise these transformations and

apply the Kalman filter to the corresponding linearised system. Many filters have been developed following the linearisation procedure [35]. In this section, one approach based on the first-order truncation of the Taylor series (see Sec. 6.2.2.1) will be presented. This technique extends the Kalman filter application to filtering problems with differentiable nonlinear functions, and therefore it is named *extended Kalman filter* [21, 43, 56]. It is clear that this method will fail if the true state is not close enough to the reference point of the expansion, i.e. when the nonlinear terms cannot be reasonably neglected.

The nonlinear filtering problem considered here is defined by the following model:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}) + G(t)\mathbf{w} \\ \mathbf{y} &= \mathbf{h}(t, \mathbf{x}) + \boldsymbol{\epsilon} \\ \mathbf{x}_0 &\sim \mathcal{N}_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, P_0).\end{aligned}\tag{6.73}$$

The nonlinear transformations $\mathbf{f}(t, \mathbf{x})$ and $\mathbf{h}(t, \mathbf{x})$ can be expanded around the reference trajectory generated by the deterministic term in the equations of motion with initial condition $\bar{\mathbf{x}}_k$:

$$\begin{aligned}\dot{\bar{\mathbf{x}}}(t) &= \mathbf{f}(t, \bar{\mathbf{x}}) \\ \bar{\mathbf{x}}(t_0) &= \bar{\mathbf{x}}_0.\end{aligned}\tag{6.74}$$

Therefore, the equations for the deviation $\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$ evolution from the reference trajectory can be approximated as:

$$\begin{aligned}\delta\dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}) - \mathbf{f}(t, \bar{\mathbf{x}}) + G(t)\mathbf{w} \\ &= \mathbf{f}(t, \bar{\mathbf{x}}) - \mathbf{f}(t, \bar{\mathbf{x}}) + \nabla_{\mathbf{x}}\mathbf{f}|_{\bar{\mathbf{x}}}\delta\mathbf{x} + \mathcal{O}(\delta\mathbf{x}^2) + G(t)\mathbf{w} \\ &\approx \nabla_{\mathbf{x}}\mathbf{f}|_{\bar{\mathbf{x}}}\delta\mathbf{x} + G(t)\mathbf{w}.\end{aligned}\tag{6.75}$$

Similarly, the observations can be defined as deviation $\delta\mathbf{y} = \mathbf{y} - \bar{\mathbf{y}}$ with respect to the deterministic measurements that would result from $\bar{\mathbf{x}}$. The resulting approximated model follows as:

$$\begin{aligned}\delta\mathbf{y} &= \mathbf{h}(t, \mathbf{x}) - \mathbf{h}(t, \bar{\mathbf{x}}) + \boldsymbol{\epsilon} \\ &= \mathbf{h}(t, \bar{\mathbf{x}}) - \mathbf{h}(t, \bar{\mathbf{x}}) + \nabla_{\mathbf{x}}\mathbf{h}|_{\bar{\mathbf{x}}}\delta\mathbf{x} + \mathcal{O}(\delta\mathbf{x}^2) + \boldsymbol{\epsilon} \\ &\approx \nabla_{\mathbf{x}}\mathbf{h}|_{\bar{\mathbf{x}}}\delta\mathbf{x} + \boldsymbol{\epsilon}.\end{aligned}\tag{6.76}$$

For both the linearised dynamics and observation model, the partial derivative Jacobian matrix is evaluated along the reference trajectory $\bar{\mathbf{x}}$. The prior distribution of the corresponding linear system follows from the linearity of the expectation operator and the fixed deterministic nature of the initial reference state:

$$\delta \mathbf{x}_0 \sim \mathcal{N}_{\delta \mathbf{x}_0}(\hat{\mathbf{x}}_0 - \bar{\mathbf{x}}_0, P_0). \quad (6.77)$$

This property on the covariance holds at any time $P_{\delta \mathbf{x}} = P_{\mathbf{x}}$ because the reference point $\hat{\mathbf{x}}$ is deterministic and fixed. The linearised system can be solved directly with the Kalman filter.

There is freedom in the choice of the reference point $\bar{\mathbf{x}}_0$, and the obvious choice is $\bar{\mathbf{x}}_0 = \hat{\mathbf{x}}_0$. In this case, from Eq. (6.77) and Eq.(6.33), it is straightforward to see that, before any observation update, $E\{\delta \mathbf{x}\} = \bar{\delta \mathbf{x}} = 0$. This is a valuable characteristic as the linearised model results accurate only for relatively small deviations. However, when an observation comes in, the update step changes the best estimate to $\hat{\mathbf{x}}_k^+$ from the reference value $\bar{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-$. In particular, when the covariance matrix of the state P_k^- is large, i.e. the estimate is not a proper measure of the distribution, the Kalman gain is large, and therefore the update step will cause the new estimate to deviate significantly from the reference state. In sequential filtering, the workaround is to re-linearise the trajectory around the new best estimate $\hat{\mathbf{x}}_k^+$ after an observation update. Indeed, if we assume that an observation is improving our knowledge of the state, then it is natural to linearise around a point supposedly closer to the true state to have smaller nonlinearity-induced errors. With this procedure, the expectation of the state deviation will be zero $\bar{\delta \mathbf{x}} = 0$ after the update step as well. The resulting numerical procedure is schematised in Algorithm 3.

Algorithm 3 Extended Kalman filter

Given the filtering model in Eq. (6.73)

- 1: Initialise $t_{k-1} = t_0$, $\bar{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_0$, $P_{k-1}^+ = P_0$, $t_k = t_1$
 - 2: **for** Observation times **do**
 - Prediction step:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)$
 - 3: Propagate reference trajectory with $\dot{\hat{\mathbf{x}}} = \mathbf{f}(t, \bar{\mathbf{x}})$
 $\bar{\mathbf{x}}_{k-1}^+ \rightarrow \bar{\mathbf{x}}_k^-$
 - 4: Propagate covariance with $\dot{P} = \nabla_{\mathbf{x}} \mathbf{f}|_{\bar{\mathbf{x}}} P_x + P_x \nabla_{\mathbf{x}} \mathbf{f}|_{\bar{\mathbf{x}}}^T + G Q G^T$
 $P_{k-1}^+ \rightarrow P_k^-$
 - Update step:** after observation $\bar{\mathbf{y}}_k$ compute $p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+)$
 - 5: Compute Kalman gain
 $K_k = P_k^- \nabla_{\mathbf{x}} \mathbf{h}|_{\bar{\mathbf{x}}}^T (\nabla_{\mathbf{x}} \mathbf{h}|_{\bar{\mathbf{x}}} P_k^- \nabla_{\mathbf{x}} \mathbf{h}|_{\bar{\mathbf{x}}}^T + R_k)^{-1}$
 - 6: Compute difference between received and predicted observations
 $\delta \bar{\mathbf{y}}_k = \bar{\mathbf{y}}_k - \mathbf{h}(t_k, \bar{\mathbf{x}}_k^-)$
 - 7: Update deviation mean and state estimate with observation information
 $\delta \bar{\mathbf{x}}_k^+ = K_k \delta \bar{\mathbf{y}}_k$, $\hat{\mathbf{x}}_k^+ = \delta \bar{\mathbf{x}}_k^+ + \bar{\mathbf{x}}_k^-$
 - 8: Update covariance with observation covariance
 $P_k^+ = P_k^- - K_k \nabla_{\mathbf{x}} \mathbf{h}|_{\bar{\mathbf{x}}} P_k^-$
 - 9: Update quantities for loop iteration
 $\bar{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_k^+$, $P_{k-1}^+ = P_k^+$, $k = k + 1$
 - 10: **end for**
-

In the prediction step, the reference trajectory is integrated exactly with the deterministic terms of the nonlinear dynamics. On the other hand, the linearised

dynamics is used to propagate the covariance information. In the same fashion, the nonlinear observation model is used to compute the difference between the received and predicted observations, while the linearised measurement model is used to update the state estimate and its corresponding covariance matrix. All the Jacobian matrices are evaluated at the reference trajectory $\{\cdot\}|_{\bar{x}}$ propagated as in step 3 of Algorithm 3. This reference state is updated after each update step to the best estimate.

As in the classic Kalman filter, the extended Kalman filter can be derived in the state transition notation. However, the procedure and resulting algorithm are similar to the classic case, and it will not be presented here. Another possible derivation is achieved by expanding directly the involved nonlinear transformations in Taylor series [51]. The same result can be generated by a statistical reasoning with least squares approach [58].

Several variants and heuristics exist to improve the basic extended Kalman filter. A solid improvement to this filter is realised with *local* iterations of the update steps 5–8 when the measurements of nonlinearity are critical [17, 29]. The iterations are called local as they are realised at a fixed time. Iterating is a regular tool for solving nonlinear problems with linear sub-steps. In the same fashion, the original update routine would involve a nonlinear measurement model, but there is no closed-form solution unless a linearisation is performed. Therefore, after the updated estimate and covariance are computed, the reference values can be updated to their values $\bar{x}_k^- = \hat{x}_k^+$ and $P_k^- = P_k^+$ and steps 5–8 repeated linearising with respect to these quantities until the changes in the optimal estimate are under a certain threshold.

In Sect. 6.2.2.1, it was shown that if the quadratic term is retained, additional terms should be included in the mean and covariance propagation (see Eqs. 6.42 and 6.43) through the nonlinear transformations, resulting in the so-called second-order extended Kalman filter. The higher order in the Taylor expansion helps to cope with the neglected system nonlinearities, at the expense of increased preliminary analytical derivations and increased computational burden. For the complete derivation, see Särkkä [51].

Plenty of variants and heuristics have been developed for generic or specific problems. In the vast literature, extensive references with a particular focus on the practical approaches are available [26, 66]. The extended Kalman filter is widely applied in navigation and orbit determination problems for space applications due to its simplicity and effectiveness [51, 58]. However, the filter may fail if the initial guess is far from the real state, a situation in which the linearised dynamics is not properly representative of the true trajectory evolution. Another drawback is that usually this method relies on the explicit derivative computation of the dynamical and measurement models, requiring rather lengthy and error-prone derivations. Therefore, generally the extended Kalman filter is not suitable for black box systems. Numerical finite-difference schemes can be implemented for the derivative computation, however resulting in worse computational performance.

6.3.3 Unscented Kalman Filter

The unscented Kalman filter (UKF) works directly with the nonlinear filtering model, i.e. without approximating the nonlinear transformations. Hence, the derivation presented for the Kalman filter and adapted for the extended version needs to be updated for nonlinear functions.

In particular, the mean and covariance of the density functions $p(\mathbf{x}_k | \mathbf{y}_{1:k-1})$ and $p(\mathbf{y}_k | \mathbf{x}_k)$ can be computed by a direct application of the unscented transform. On the other hand, Eq. (6.60) should be rewritten for a general nonlinear observation relationship. In a generic notation, for $\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\epsilon}$ (neglecting the explicit dependency on time), $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, P)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, R)$, under the application of the unscented transform with posterior Gaussian, it holds that [51]:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}_{\mathbf{x}, \mathbf{y}} \left(\begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{y}} \end{pmatrix}, \begin{pmatrix} P & C \\ C^T & S \end{pmatrix} \right), \quad (6.78)$$

where $\hat{\mathbf{y}}$ is the expected value of the observations, S the observation's covariance matrix and C the cross covariance between state and observations. These quantities are computed using the unscented transformation samples \mathbf{x}_i and $\mathbf{y}_i = \mathbf{h}(\mathbf{x}_i)$. Specifically, $\hat{\mathbf{y}}$ is computed directly using Eq. (6.44), whereas S is obtained by Eq. (6.45) with the addition of the additive noise covariance as:

$$S \approx \sum_{i=0}^{N_\sigma} w_i (\mathbf{y}_i - \hat{\mathbf{y}})(\mathbf{y}_i - \hat{\mathbf{y}})^T + R. \quad (6.79)$$

The cross covariance matrix is computed by the samples and corresponding responses deviations from the reference value:

$$C \approx \sum_{i=0}^{N_\sigma} w_i (\mathbf{x}_i - \hat{\mathbf{x}})(\mathbf{y}_i - \hat{\mathbf{y}})^T. \quad (6.80)$$

Adapting Eq. (6.78) for conditional probabilities in the sequential filtering framework, comparing it with Eq. (6.60) and repeating the same marginalisation procedure as in Sect. 6.3.1, the Kalman gain can be equivalently defined for the unscented Kalman filter as [51, 55]:

$$K_k = C_k S_k^{-1}. \quad (6.81)$$

With these new definitions, the update step of the unscented Kalman filter is reformulated as:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_k) \quad (6.82)$$

$$P_k^+ = P_k^- - K_k S_k K_k^T. \quad (6.83)$$

The algorithm of the unscented Kalman filter therefore follows as schematised in Algorithm 4, where the formulation with $2N_x + 1$ sigma points is used.

Algorithm 4 Unscented Kalman filter

Given the filtering model in Eq. (6.73)

1: Initialise $t_{k-1} = t_0$, $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_0$, $P_{k-1}^+ = P_0$, $t_k = t_1$

2: **for** Observation times **do**

Prediction step: compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)$

3: Select sigma points and relative weights from Eq. (6.47) (or modified $w_i^{(m)}$, $w_i^{(c)}$)

$$\begin{cases} \mathbf{x}_0(t_{k-1}) &= \hat{\mathbf{x}}_{k-1}^+ \\ \mathbf{x}_i(t_{k-1}) &= \hat{\mathbf{x}}_{k-1}^+ + \sqrt{(N_x + \lambda) P_{k-1}^{+(i)}} & \text{for } i = 1, \dots, N_x \\ \mathbf{x}_i(t_{k-1}) &= \hat{\mathbf{x}}_{k-1}^+ - \sqrt{(N_x + \lambda) P_{k-1}^{+(i-N_x)}} & \text{for } i = N_x + 1, \dots, 2N_x \end{cases}$$

4: Propagate samples with nonlinear dynamics $\dot{\mathbf{x}}_i = \mathbf{f}(t, \mathbf{x}_i)$ for $i = 0, \dots, 2N_x$
 $\mathbf{x}_i(t_{k-1}) \rightarrow \mathbf{x}_i(t_k)$

5: Compute predicted state mean and covariance

$$\hat{\mathbf{x}}_k^- = \sum_{i=0}^{2N_x} w_i^{(m)} \mathbf{x}_i(t_k), \quad P_k^- = \sum_{i=0}^{2N_x} w_i^{(c)} (\mathbf{x}_i(t_k) - \hat{\mathbf{x}}_k^-)(\mathbf{x}_i(t_k) - \hat{\mathbf{x}}_k^-)^T$$

Update step: after observation $\bar{\mathbf{y}}_k$ compute $p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+)$

6: Select new sigma points

$$\begin{cases} \mathbf{x}_0(t_k) &= \hat{\mathbf{x}}_k^- \\ \mathbf{x}_i(t_k) &= \hat{\mathbf{x}}_k^- + \sqrt{(N_x + \lambda) P_k^{-(i)}} & \text{for } i = 1, \dots, N_x \\ \mathbf{x}_i(t_k) &= \hat{\mathbf{x}}_k^- - \sqrt{(N_x + \lambda) P_k^{-(i-N_x)}} & \text{for } i = N_x + 1, \dots, 2N_x \end{cases}$$

7: Propagate samples with nonlinear observation model $\mathbf{y}_i = \mathbf{h}(t, \mathbf{x}_i)$ for $i = 0, \dots, 2N_x$
 $\mathbf{x}_i(t_k) \rightarrow \mathbf{y}_i(t_k)$

8: Compute predicted observation mean, covariance and state observation cross covariance

$$\hat{\mathbf{y}}_k = \sum_{i=0}^{2N_x} w_i^{(m)} \mathbf{y}_i(t_k),$$

$$S_k = \sum_{i=0}^{2N_x} w_i^{(c)} (\mathbf{y}_i(t_k) - \hat{\mathbf{y}}_k)(\mathbf{y}_i(t_k) - \hat{\mathbf{y}}_k)^T, \quad C_k = \sum_{i=0}^{2N_x} w_i^{(c)} (\mathbf{x}_i(t_k) - \hat{\mathbf{x}}_k^-)(\mathbf{y}_i(t_k) - \hat{\mathbf{y}}_k)^T$$

9: Compute Kalman gain

$$K_k = C_k S_k^{-1}$$

10: Update mean with observation information

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k (\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_k)$$

11: Update covariance with observation covariance

$$P_k^+ = P_k^- - K_k S_k K_k^T$$

12: Update quantities for loop iteration

$$\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_k^+, \quad P_{k-1}^+ = P_k^+, \quad k = k + 1$$

13: **end for**

Since the approximation is performed on the distributions directly, the unscented Kalman filter does not require differentiability or derivative knowledge of the nonlinear transformations. Therefore this method is suitable for black box implementation. In general, the unscented transformation is more accurate in propagating the density mean and covariance through a nonlinear function than Taylor-based linearisation for a comparable computational cost [34, 55]. Julier and Uhlmann [31]

claimed that overall the unscented Kalman filter is a more direct generalisation of the Kalman filter rather than the extended Taylor series one. Indeed, the same reasoning addressed at the beginning of the section on the definition of linear filtering arises. There is no theoretical need of linearity on the dynamical and measurement models to approximate the posterior as a normal distribution, and the unscented transform is a powerful tool to achieve such approximation by only evaluating a set of Gaussian moment equations for the selected propagated samples [26].

As for the other filters, plenty of variants, heuristics and generalisations were developed for the unscented Kalman filter. Among them, some involve the use of different numbers of sigma points resulting in higher-order techniques, as already outlined in Sect. 6.2.2.2. An alternative filter can also be formulated to account for a non-additive noise contribution, by augmenting the state vector with process and observation noises before using the unscented transformation [34, 51]. For the difference in the derivations by state augmentation, see Wu et al. [64]. Extensive references for the original unscented filter and its numerous extensions can be found in the literature [59].

Although historically developed and extensively employed with Gaussian priors and posteriors, which provide a clear and simple result, it is worth to recall that there is no need to rely on the Gaussianity assumption at all in the unscented transformation.

6.3.4 Gaussian Filter Framework

The current section dealt entirely with filtering techniques which, directly or indirectly, approximate the posterior distribution $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ as Gaussian. The extended Kalman filter constructs indirectly a Gaussian posterior by linearising the nonlinear transformations, therefore ensuring the conservation of the distribution's Gaussian nature (see Sec. 6.2.1). The unscented Kalman filter directly fits a Gaussian distribution to the propagated samples by matching the first two moments of the resulting discrete density.

The latter idea was shown to be a particular case of a general framework for *Gaussian assumed density filter* [28, 42, 51, 65]. The goal is again to approximate the posterior density $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ as Gaussian, regardless of the nonlinearity properties of the dynamical and measurement models.

For a general nonlinear transformation $\mathbf{z} = \mathbf{g}(\mathbf{x})$, with $\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, P_x)$, the moment matching approximation is constructed as in Eq. (6.78). Now, the moments are computed by the expectation operator. Explicitly:

$$p(\mathbf{x}, \mathbf{z}) = \mathcal{N}_{\mathbf{x}, \mathbf{z}} \left(\begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}} \end{pmatrix}, \begin{pmatrix} P & C \\ C^T & S \end{pmatrix} \right),$$

where now:

$$\begin{aligned}
\hat{\mathbf{z}} &= \int \mathbf{g}(\mathbf{x}) \mathcal{N}_{\mathbf{x}}(\hat{\mathbf{x}}, P_x) d\mathbf{x} \\
S &= \int (\mathbf{g}(\mathbf{x}) - \hat{\mathbf{z}})(\mathbf{g}(\mathbf{x}) - \hat{\mathbf{z}})^T \mathcal{N}_{\mathbf{x}}(\hat{\mathbf{x}}, P_x) d\mathbf{x} \\
C &= \int (\mathbf{x} - \hat{\mathbf{x}})(\mathbf{g}(\mathbf{x}) - \hat{\mathbf{z}})^T \mathcal{N}_{\mathbf{x}}(\hat{\mathbf{x}}, P_x) d\mathbf{x}.
\end{aligned} \tag{6.84}$$

From the moment matching formulation, a Gaussian can be fit to all the probability densities involved in the prediction and update step by using the definition of expectation. By defining the integral equations of motion as

$$\mathbf{x}(t_k) = \mathbf{F}(\mathbf{x}_{k-1}) \triangleq \int_{t_{k-1}}^{t_k} \mathbf{f}(t, \mathbf{x}) dt + \mathbf{x}(t_{k-1}). \tag{6.85}$$

the general algorithm for the Gaussian filter is schematised in Algorithm 5, using the same form of update equations as presented for the UKF [51].

Algorithm 5 Gaussian filter

Given the filtering model in Eq. (6.73)

- 1: Initialise $t_{k-1} = t_0$, $\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_0$, $P_{k-1}^+ = P_0$, $t_k = t_1$
 - 2: **for** Observation times **do**
 - 3: **Prediction step:** compute $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^-, P_k^-)$
 - 4: Compute mean estimate
$$\hat{\mathbf{x}}_k^- = \int \mathbf{F}(\mathbf{x}_{k-1}) \mathcal{N}(\hat{\mathbf{x}}_{k-1}^+, P_{k-1}^+) d\mathbf{x}_{k-1}$$
 - 4: Propagate covariance matrix
$$P_k^- = \int (\mathbf{F}(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_k^-)(\mathbf{F}(\mathbf{x}_{k-1}) - \hat{\mathbf{x}}_k^-)^T \mathcal{N}(\hat{\mathbf{x}}_{k-1}^+, P_{k-1}^+) d\mathbf{x}_{k-1}$$
 - 5: **Update step:** after observation $\bar{\mathbf{y}}_k$ compute $p(\mathbf{x}_k | \mathbf{y}_{1:k}) = \mathcal{N}_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+, P_k^+)$
 - 5: Compute predicted observation mean, covariance and state observation cross covariances
$$\hat{\mathbf{y}}_k = \int \mathbf{h}(t_k, \mathbf{x}_k) \mathcal{N}(\hat{\mathbf{x}}_k^-, P_k^-) d\mathbf{x}_k$$

$$S_k = \int (\mathbf{h}(t_k, \mathbf{x}_k) - \hat{\mathbf{y}}_k)(\mathbf{h}(t_k, \mathbf{x}_k) - \hat{\mathbf{y}}_k)^T \mathcal{N}(\hat{\mathbf{x}}_k^-, P_k^-) d\mathbf{x}_k$$

$$C_k = \int (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{h}(t_k, \mathbf{x}_k) - \hat{\mathbf{y}}_k)^T \mathcal{N}(\hat{\mathbf{x}}_k^-, P_k^-) d\mathbf{x}_k$$
 - 6: Compute Kalman gain
$$K_k = C_k S_k^{-1}$$
 - 7: Update mean with observation information
$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\bar{\mathbf{y}}_k - \hat{\mathbf{y}}_k)$$
 - 8: Update covariance with observation covariance
$$P_k^+ = P_k^- - K_k S_k K_k^T$$
 - 9: Update quantities for loop iteration
$$\hat{\mathbf{x}}_{k-1}^+ = \hat{\mathbf{x}}_k^+, P_{k-1}^+ = P_k^+, k = k + 1$$
 - 10: **end for**
-

All the methods developed previously can be rederived from this general form depending on how the integrals above are solved. Indeed, this is the generalisation for general nonlinear functions of Eq. (6.60), used to derive the KF. The EKF is obtained by approximating $\mathbf{g}(\mathbf{x})$ at the first order in Eq. (6.84), which then can be

solved analytically. In addition, it can be shown that Eqs. (6.79)–(6.80), used in the UKF derivation, are the result of a Gauss-Hermite cubature rule application to Eq. (6.84) [28].

Recently, numerous novel filtering techniques have arisen from this general moment matching formulation, as it allows to use any approximation rule for the integral computation. Among the deterministic methods, Gauss-Hermite quadrature and spherical cubature are efficient schemes [2, 28, 51, 65]. Also nondeterministic methods can be used, such as the often employed Monte Carlo family sampling techniques.

6.3.5 Particle Filter

The nonlinear model in Eq. (6.73), or its non-additive noise counterpart, can be used to compute the conditional probabilities $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and $p(\mathbf{y}_k|\mathbf{x}_k)$, whose distribution depends on the assumed probability density function of the process and observation noises. Therefore, these quantities are assumed to be given in the problem formulation tackled in this section [51].

The particle filter is a state estimation technique based on sequential Monte Carlo methods (see Sect. 6.2.2.3). As the name implies, it relies on a set of weighted random particles to approximate the posterior distribution [3, 24]:

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_i w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \quad (6.86)$$

where $\delta(\cdot)$ is the Dirac delta function. From this distribution, the expectation of a generic function, and therefore its moments, can be computed by the weighted sum [51]:

$$E\{\mathbf{g}(\mathbf{x}_k)|\mathbf{y}_{1:k}\} \approx \sum_i w_k^{(i)} \mathbf{g}(\mathbf{x}_k^{(i)}). \quad (6.87)$$

The sample and weights are computed using the sequential importance sampling technique adapted to account for the observations. With the help of the importance distribution π , the weights at step k are defined by:

$$\begin{aligned}
w_k^{(i)} &= \frac{p(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k})} = \frac{p(\mathbf{y}_k | \mathbf{x}_{0:k}^{(i)}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k-1}) / p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})} \\
&\approx \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})} \\
&= \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})} \\
&= w_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})}, \tag{6.88}
\end{aligned}$$

where the Markov properties (6.7)–(6.8) have been used. The normalisation quantity $p(\mathbf{y}_k | \mathbf{y}_{1:k-1})$ has disappeared because when the weights are normalised to sum to unity, $w_k^{(i)*} = w_k^{(i)} / \sum_j w_k^{(j)}$, it cancels out regardless. From the problem formulation in Eq.(6.73), and because $\pi(\cdot)$ should be chosen to be simple to sample from, it is straightforward to evaluate this weight update equation. As in Sect. 6.2.2.3, the importance sampling was chosen to decompose as:

$$\pi(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k}) = \pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k}) \pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1}). \tag{6.89}$$

This choice is key in the sequential algorithm, as the sample $\mathbf{x}_{0:k}^{(i)}$ from $\pi(\mathbf{x}_{0:k}^{(i)} | \mathbf{y}_{1:k})$ can be obtained by simply augmenting $\mathbf{x}_{0:k-1}^{(i)}$ from $\pi(\mathbf{x}_{0:k-1}^{(i)} | \mathbf{y}_{1:k-1})$ with $\mathbf{x}_k^{(i)}$ from $\pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$, avoiding therefore to sample the full joint distribution [3].

With the developed sequential framework, the general scheme of a particle filter algorithm is given in Algorithm 6. There are infinite ways to choose the importance

Algorithm 6 Particle filter (no resampling)

Given the filtering model in Eq. (6.73)

- 1: Draw N particles $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, with equal weights $w_0^{(i)} = 1/N$
 - 2: **for** $k=1$: Observation times **do**
 - 3: Sample new particles $\mathbf{x}_k^{(i)}$ and augment vector $\mathbf{x}_{0:k}^{(i)} = [\mathbf{x}_{0:k-1}, \mathbf{x}_k^{(i)}]$
 $\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$
 - 4: Compute corresponding weights $w_k^{(i)}$ and normalise to $w_k^{(i)*}$
 $w_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}) / \pi(\mathbf{x}_k^{(i)} | \mathbf{x}_{0:k-1}^{(i)}, \mathbf{y}_{1:k})$
 $w_k^{(i)*} = w_k^{(i)} / \sum_j w_k^{(j)}$
 - 5: **end for**
-

distribution. Obviously, good properties are the simplicity to draw samples from it and the ease to evaluate the probability density associated to a particle.

The *Bootstrap filter* is a particle filter which employs the transitional density as importance distribution [24]:

$$\pi = p(\mathbf{x}_k | \mathbf{x}_{k-1}) . \quad (6.90)$$

This particular choice leads to the simplification of the weight update equation to:

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) . \quad (6.91)$$

The resulting algorithm, the first particle filter ever, is simple, intuitive and modular. Indeed, the samples are simple to draw from the transitional density, and the weight update requires the evaluation of the observation's conditional density, given by problem formulation. On the other hand, it draws samples according to the dynamical information only. Hence, when there is little overlap between the predicted and the observation distributions, most of the particles will be associated to small importance weights, and the posterior distribution's approximation will be dominated by a very limited number of particles with large weights [48]. Since each particle requires the same amount of computational load, this filter implementation is often inefficient as it requires a high number of particles for accurate approximations.

It is clear how a trade-off between the resemblance of the importance distribution to the true posterior and the computational efficiency is the key of particle filters. Plenty of research has been, and still is, focused on the selection of optimal importance distributions. As a general rule, it is advantageous to retain the conditionality on the last measurement [48].

One alternative which minimises the variance of the importance weights is [19]:

$$\pi = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k) . \quad (6.92)$$

This importance density leads to the weight update equation:

$$w_{k-1}^{(i)} = w_{k-1}^{(i)} p(\mathbf{y}_k | \mathbf{x}_{k-1}^{(i)}) . \quad (6.93)$$

However, both the equations cannot be directly used. When this is the case, local linearisation techniques, e.g. EKF or UKF, can be employed to create suitable importance distribution [19, 51, 61].

As introduced in Sect. 6.2.2.3, and discussed for the Bootstrap filter, one issue often encountered is the weight degeneracy as a result of sampling from an inappropriate importance distribution. Therefore, another way to improve computational efficiency is to introduce resampling techniques, which remove low-weighted particles and replace them with duplicates of the high-weighted samples. As it is often a matter of heuristics when and how this resampling step should be

performed, numerous alternatives have been studied in the literature. The *Bootstrap filter* performs a *resampling step* after each observation update, which substitutes the particles with a new set of particles according to the current discrete density in Eq. (6.86) and re-initialises the weights to $w_k^{(i)} = 1/N$. Alternatively, the resampling step can be performed after n weight updates. Another alternative requires the introduction of a check step in which the variance of the weights is assessed, leading to the so-called adaptive resampling. If this variance becomes too high, or equivalently its inverse too low, the particles are resampled [39]. In general, this approach helps to better distribute the samples in the zones where the weights are higher, and therefore the resampling step is always present in particle filters [51].

However, along with its advantages, the resampling step introduces an undesired phenomenon called *sample impoverishment* [49]. Indeed, the resampling technique replicates, possibly several times, particles associated with high weights after a filter iteration. Then, these samples are propagated via the importance distribution, and they are supposed to diversify as a result of the process noise. However, if the process noise is small, the same initial particles will end up to be close after propagation. Eventually, in degenerate cases, all the particles will collapse to the same point [55]. Several techniques exist to mitigate this issue: *roughening*, which adds random noise to the particle just after the resampling process [20, 24]; *prior editing*, using roughening on the prior samples with small weights [24, 55]; *regularized particle filtering*, which performs resampling from a continuous approximated auxiliary density function [20, 49]; *Markov chain Monte Carlo resampling* [22, 49] and *auxiliary particle filtering* [46].

The particle filter suffers from the so-called curse of dimensionality, as several studies have shown that the number of particles needed for a successful filtering process scales exponentially with the state dimension [5, 57, 62]. When some components of the state vector follow a linear evolution and they are Gaussian, while the others are non-Gaussian, the computational burden can be reduced by evaluating part of the filtering equations analytically, while the rest still require sampling techniques [26, 51]. The resulting algorithm is called Rao-Blackwellized particle filter [1, 11, 45].

Although relatively recent, there exists an extensive literature on particle filters, its variants and associated heuristics. Mainly, this topic can be found in the literature focused on sequential Monte Carlo methods in general [20, 38], or on filtering techniques in particular [10, 49, 55].

6.4 Conclusions

State estimation theory is of crucial importance for a great variety of fields. In this framework, the time-varying state of a hidden dynamical system is sought by combining uncertain evolution knowledge with noisy observations.

This chapter introduced both the fundamental concepts of state estimation in general, and filtering theory in particular, through its probabilistic development, and practical techniques for computing its solution.

In Sect. 6.1, the general mathematical statement was introduced with the building blocks which are necessary for the filtering problem discussed in this chapter: time-continuous dynamical equations, an observation model and a known initial distribution of the state. Within the class of state estimation, the main focus was filtering theory, which aims at computing the state distribution at the time of the last received observation. Hence, filtering is appropriate for real-time applications. The inference step needed to combine dynamical and measurement information was solved by Bayes' rule. Its application to the current setting was presented, and two mathematically equivalent update rules were derived. One of them suits a sequential scheme convenient for real-time applications (sequential filtering), while the other processes a whole set of observations at once (batch processor). As the chapter focused on the former, key importance was given on analytical and numerical techniques to compute the corresponding update step. The section ended with a discussion on which estimate should be used as representative of the state probability conditional distribution. A general method to compute optimal statistical estimators via loss functions was presented. Among the alternatives, specific loss functions allow to select the conditional distribution mean, mode, median, etc.

For any filtering algorithm, one necessary step is to be able to describe, or approximate, how the state distribution evolves through the dynamical equations and measurement model. In Sect. 6.2, the methods for propagating probability distributions through a transformation were presented. Initially, the exact solution for linear dynamics, linear observation model and zero-mean Gaussian noises was presented: if the input random variable is normally distributed, the transformed variable is still Gaussian with mean and variance analytically computed. Moreover, the case of a generic nonlinear function was considered and methods to approximate the transformed distribution introduced. Specifically, the Taylor expansion and the unscented transform approximate, respectively, the transformation and the posterior to have a final normal distribution, whereas Monte Carlo methods are able to describe generic posteriors using sampling-based discrete distributions.

Lastly, in Sect. 6.3, practical algorithms to compute or approximate the filtering solution were derived, described and schematised. Specifically, when linear dynamics, linear observation model and Gaussian prior and noises are considered, the Kalman filter is the closed-form solution of the filtering problem. However, most state estimation problems involve nonlinear transformations, and a general analytical solution is not available. When the Gaussian assumption (or approximation) is retained, a family of methods exists to obtain a normal posterior distribution. In detail, the Taylor expansion approximation results in the extended Kalman filter, while the unscented transform is the basis for the unscented Kalman filter. To conclude the family of Gaussian filtering methods, the general framework of

Gaussian filters, which approximate the posterior as Gaussian via moment matching approximations, was sketched. Lastly, the sampling-based particle filter was derived as a general practical method to compute the filtering solution when the assumptions of the previous methods are too restrictive, e.g. when distributions other than Gaussian are involved.

References

1. H. Akashi, H. Kumamoto, Random sampling approach to state estimation in switching environments. *Automatica* **13**(4), 429–434 (1977)
2. I. Arasaratnam, S. Haykin, Cubature Kalman filters. *Trans. Autom. Control.* **54**(6), 1254–1269 (2009)
3. M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. Signal Proc.* **50**(2), 174–188 (2002)
4. R.H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised edn. (American Institute of Aeronautics and Astronautics, Reston, 1999)
5. A. Beskos, D. Crisan, A. Jasra, K. Kamatani, Y. Zhou, A stable particle filter in high-dimensions (2014). Preprint arXiv:1412.3501
6. AT. Bharucha-Reid, *Elements of the Theory of Markov Processes and Their Applications* (McGraw-Hill, New York, 1960)
7. G.J. Bierman, *Factorization Methods for Discrete Sequential Estimation* (Dover, Illinois, 2006)
8. S. Brooks, A. Gelman, G.L. Jones, X.L. Meng, *Handbook of Markov Chain Monte Carlo* (Chapman & Hall/CRC, Boca Raton, 2011)
9. R. Bucy, P. Joseph, *Filtering for Stochastic Processes* (Wiley, Hoboken, 1968)
10. J.V. Candy, *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*, 2nd edn. (Wiley, Hoboken, 2016). <https://doi.org/10.1002/9781119125495>
11. C. Casella, C.P. Rober, Rao-Blackwellisation of sampling schemes. *Biometrika* **83**(1), 81–94 (1996)
12. S. Chakravorty, M. Kumar, P. Singla, A quasi-Gaussian Kalman filter, in *American Control Conference*, Minneapolis (2006). <https://doi.org/10.1109/ACC.2006.1655484>
13. S. Challa, Y. Bar-Shalom, Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions. *IEEE Trans. Aerosp. Electron. Syst.* **36**(1), 309–315 (2000). <https://doi.org/10.1109/7.826335>
14. S. Challa, Y. Bar-Shalom, V. Krishnamurthy, Nonlinear filtering via generalized Edgeworth series and Gauss-Hermite quadrature. *IEEE Trans. Signal Proc.* **48**(6), 1816–1820 (2000). <https://doi.org/10.1109/78.845944>
15. S. Challa, F.A. Faruqi, Application of Chebechev's inequality theorem in the design of optimal non-linear filters, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3 (1998). <https://doi.org/10.1109/ICASSP.1998.681678>
16. H. Cox, On the Estimation of state variables and parameters for noisy dynamic systems. *IEEE Trans. Autom. Control* **9**(1), 5–12 (1964). <https://doi.org/10.1109/TAC.1964.1105635>
17. W.F. Denham, S. Pines, Sequential estimation when measurement function nonlinearity is comparable to measurement error. *AIAA J.* **4**(6), 1071–1076 (1966). <https://doi.org/10.2514/3.3606>
18. A. Doucet, W. Xiaodong, Monte Carlo methods for signal processing: a review in the statistical signal processing context. *IEEE Signal Proc. Mag.* **22**(6), 152–170 (2005). <https://doi.org/10.1109/MSP.2005.1550195>
19. A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for bayesian filtering. *Stat. Comput.* **10**(3), 197–208 (2000). <https://doi.org/10.1023/A:1008935410038>

20. A. Doucet, N. De Freitas, N. Gordon, *Sequential Monte Carlo Methods in Practice*, 1st edn. (Springer, New York, 2001). <https://doi.org/10.1007/978-1-4757-3437-9>
21. A. Gelb, *Applied Optimal Estimation* (MIT Press, Cambridge, 1974)
22. W. Gilks, C. Berzuini, Following a moving target—Monte Carlo inference for dynamic Bayesian models. *J. Royal Stat. Soc.* **63**, 127–146 (2002). <https://doi.org/10.1111/1467-9868.00280>
23. W. Gilks, S. Richardson, D. Spiegelhalter, *Markov Chain Monte Carlo in Practice* (Chapman and Hall/CRC, Boca Raton, 1996)
24. N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F (Radar and Signal Processing)* **140**(2), 107–113 (1993). <https://doi.org/10.1049/ip-f-2.1993.0015>
25. M.S. Grewal, A.P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 4th edn. (Wiley-IEEE Press, Hoboken, 2014)
26. A.H. Haug, *Bayesian Estimation and Tracking: A Practical Guide* (Wiley, Hoboken, 2012)
27. G. Hendebey, F. Gustafsson, On nonlinear transformations of stochastic variables and its application to nonlinear filtering, in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing* (2008). <https://doi.org/10.1109/ICASSP.2008.4518435>
28. K. Ito, K. Xiong, Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control.* **45**(5), 910–927 (2000)
29. A.H. Jazwinski, *Stochastic Processes and Filtering Theory*, ed. by R. Bellman. Mathematics in Science and Engineering, vol. 64, 1st edn. (Academic, New York, 1970)
30. S. Julier, The spherical simplex unscented transformation, in *Proceedings of the American Control Conference*, vol. 3 (2003). <https://doi.org/10.1109/ACC.2003.1243439>
31. S. Julier, J.K. Uhlmann, A general method for approximating nonlinear transformations of probability distributions. Technical Report Robotics Research Group, University of Oxford (1996)
32. S. Julier, J.K. Uhlmann, A consistent, debiased method for converting between polar and Cartesian coordinate systems, in *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls* (1997)
33. S. Julier, J.K. Uhlmann, Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations, in *Proceedings of the American Control Conference*, vol. 2 (2002). <https://doi.org/10.1109/ACC.2002.1023128>
34. S. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation. *Proc. IEEE* **92**(3), 401–422 (2004). <https://doi.org/10.1109/JPROC.2003.823141>
35. T. Kailath, A.H. Sayed, B. Hassibi, *Linear Estimation*, 1st edn. (Prentice Hall, Upper Saddle River, 2000)
36. R.E. Kalman, A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**(1), 35–45 (1960). <https://doi.org/10.1115/1.3662552>
37. D. Knill, W. Richards, *Perception as Bayesian Inference*, 1st edn. (Cambridge University Press, Cambridge, 1996). <https://doi.org/10.1017/CBO9780511984037>
38. J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, 1st edn. (Springer, New York, 2004)
39. J.S. Liu, R. Chen, Blind deconvolution via sequential imputations. *J. Amer. Stat. Assoc.* **90**(430), 567–576 (1995)
40. M. Majji, J.L. Junkins, J.D. Turner, A perturbation method for estimation of dynamic systems. *Nonlinear Dyn.* **60**(3), 303–325 (2010). <https://doi.org/10.1007/s11071-009-9597-6>
41. A.M. Mathai, S.B. Provost, *Quadratic Forms in Random Variables* (CRC Press, Boca Raton, 1992)
42. P. Maybeck, *Stochastic Models, Estimation and Control*, vol. 2 (Academic, Cambridge, 1982)
43. B.A. McElhoe, An assessment of the navigation and course corrections for a manned flyby of mars or venus. *IEEE Trans. Aerosp. Electron. Syst.* **AES-2**, 613–623 (1966)
44. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087 (1953). <https://doi.org/10.1063/1.1699114>

45. K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian networks, in *Sequential Monte Carlo Methods in Practice* (Springer, Berlin, 2001)
46. M.K. Pitt, N. Shephard, Filtering via simulation: auxiliary particle filters. *J. Amer. Stat. Asso.* **94**(446), 590–599 (1999)
47. H. Risken, *The Fokker-Planck Equation: Methods of Solution and Applications*, 2nd edn. (Springer, Berlin, 1989)
48. B. Ristic, *Particle Filters for Random Set Models* (Springer, New York, 2013) <https://doi.org/10.1007/978-1-4614-6316-0>
49. B. Ristic, S. Arulampalam, N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications* (Artech House, Norwood, 2004)
50. T.P. Sapsis, G. Athanassoulis, New partial differential equations governing the response-excitation joint probability distributions of nonlinear systems under general stochastic excitation. *Probab. Eng. Mech.* **23**(2–3), 289–306 (2008). <https://doi.org/10.1016/j.probingmech.2007.12.028>
51. S. Sarkka, *Bayesian Filtering and Smoothing*, 1st edn. (Cambridge University Press, New York, 2013)
52. F.H. Schlee, C.J. Standish, N.F. Toda, Divergence in the Kalman filter. *AIAA J.* **5**(6), 1114–1120 (1967)
53. S.N. Sharma, A Kolmogorov-Fokker-Planck approach for a stochastic Duffing-van der Pol system. *Differ. Equ. Dyn. Syst.* **16**(4), 351–377 (2008). <https://doi.org/10.1007/s12591-008-0019-x>
54. A.N. Shiryayev, *Probability*, 2nd edn. (Springer, Berlin, 1996)
55. D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, 1st edn. (Wiley-Interscience, Hoboken, 2006)
56. G.L. Smith, S.F. Schmidt, L.A. McGee, Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. NASA Technical Report (1962)
57. C. Snyder, T. Bengtsson, P. Bickel, J. Anderson, Obstacles to high-dimensional particle filtering. *Monthly Weather Rev.* **136**(12), 4629–4640 (2008)
58. B.D. Tapley, B.E. Schutz, G.H. Born, *Statistical Orbit Determination*, 1st edn. (Elsevier Academic, San Diego, 2004)
59. R. Van der Merwe, Sigma-point kalman filters for probabilistic inference in dynamic state-space models. Doctoral Thesis at Oregon Health & Science University (2004)
60. R. Van der Merwe, E.A. Wan, *Kalman Filtering and Neural Networks* (Wiley, Hoboken, 2002)
61. R. Van Der Merwe, A. Doucet, N. De Freitas, E.A. Wan, The unscented particle filter, in *Advances in Neural Information Processing Systems* (2001)
62. P.J. van Leeuwen, Aspects of particle filtering in high-dimensional spaces, in *Dynamic Data-Driven Environmental Systems Science* (Springer, Cham, 2015)
63. E.A. Wan, R. Van der Merwe, The unscented Kalman filter for nonlinear estimation, in *Proceedings of IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium* (2000). <https://doi.org/10.1109/ASSPCC.2000.882463>
64. Y. Wu, D. Hu, M. Wu, X. Hu, Unscented Kalman filtering for additive noise case: augmented versus nonaugmented. *IEEE Signal Proc. Lett.* **12**(5), 357–360 (2005)
65. Y. Wu, D. Hu, M. Wu, X. Hu, A numerical-integration perspective on Gaussian filters. *IEEE Trans. Signal Proc.* **54**(8), 2910–2921 (2006)
66. P. Zarchan, H. Musoff, F.K. Lu, *Fundamentals of Kalman Filtering: A Practical Approach*, 3rd edn. (American Institute of Aeronautics & Astronautics, Reston, 2009)

Chapter 7

Introduction to Optimisation



Annalisa Riccardi, Edmondo Minisci, Kerem Akartunali, Cristian Greco, Naomi Rutledge, Alexander Kershaw, and Aymen Hashim

Abstract This chapter gives a brief introduction to the formulation of optimisation problems and solving algorithms. After mentioning the different classes of problems, such as continuous/discrete, local/global and single-/multi-objective, and introducing some of the useful terminology, the chapter is split in two main parts: (1) formulations and algorithms for continuous problems, including optimal control, and (2) formulations and algorithms for integer and mixed-integer problems. Both sections first consider standard deterministic methods that have been derived starting by optimality criteria, then more recent heuristics derived by experience and sometimes inspired by nature. This gives the basis to better read and understand some of the following chapters on more advanced topics.

Keywords Optimisation · Optimal control · Network optimisation · Multi-objective · Continuous variables · Combinatorial variables

7.1 Introduction

Optimisation, derived from the Latin word *optimus* meaning ‘the best’, is the general name used to characterise the process of finding the best possible solution for a problem given a measure of ‘goodness’. This is, for example, the problem of finding the shortest or fastest route between two points or the best investment in the stock market that minimises risk and maximises return.

People have been optimising since the beginning of the humankind era, but the roots for modern-day mathematical and engineering optimisation can be traced to the Second World War, where optimisation processes were formalised, implemented and applied to practical operational problems. The term operational research (OR)

A. Riccardi (✉) · E. Minisci · K. Akartunali · C. Greco · N. Rutledge · A. Kershaw · A. Hashim
University of Strathclyde, Glasgow, UK
e-mail: annalisa.riccardi@strath.ac.uk; edmondo.minisci@strath.ac.uk;
kerem.akartunali@strath.ac.uk; c.greco@strath.ac.uk; naomi.rutledge.2013@uni.strath.ac.uk;
alexander.kershaw.2013@uni.strath.ac.uk; aymen.hashim.2013@uni.strath.ac.uk

originated from the activities performed by teams of multidisciplinary experts in the armed forces that were using advanced analytical methods to devise better decisions. Applications in the service industries did not begin until the mid-1960s, where the knowledge generated during the war was applied to logistic-related problems.

The term ‘programming’ is often used in relation to optimisation: mathematical programming, linear programming, non-linear programming, mixed-integer programming, etc. In principal, the original use of the word ‘programming’ has little to do with modern-day computer programming. Before the days of computing, a set of values which represented a solution to a problem was referred to as a programme. Nowadays, software is programmed to find a set of optimal values (or ‘programme’) for your problem. The intention of optimisation in modern-day programming is to maximise or minimise an objective function (performance measure indicator) with respect to a set of variables (optimisation variables) subject to one or more constraints. Modern mathematical optimisation can be used in a wide array of fields and disciplines, ranging from the design of aircrafts, the planning of routes and schedules, to the design of a control profile for an operating machine. In any optimisation problem, there are formulation and programming challenges that must be overcome to find an optimal solution. Some of them are discussed in the next section.

7.1.1 Solving an Optimisation Problem

There are three main challenges, or steps, to be addressed when facing a general optimisation problem: problem formulation, problem characteristics and algorithm selection.

- Problem formulation: the problem, originally described in general terms, needs to be translated into its mathematical formulation, including the identification of the set of optimisation variables and constant problem parameters, definition of objectives and constraints.
- Problem characteristics: the dimension of the design vector space (number of optimisation variables) and its nature (continuous or discrete), dimension of the objectives and constraints space (number of performance measures and constraints functions), their degree of non-linearity, their smoothness, their landscape as well as their computational cost.
- Algorithm selection: from the pool of available algorithms the most suitable algorithm needs to be selected to solve the formulated problem.

Without loss of generalisation we can restrict ourselves to discuss only the case of minimisation: find $\mathbf{x}^* \in \Omega \subseteq \mathbb{R}^{n_x}$

$$f(\mathbf{x}^*) = \min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

subject to $c(\mathbf{x}) \leq 0$,

where $f : \Omega \rightarrow \mathbb{R}^{n_{\text{obj}}}$ is the objective function and $c : \Omega \rightarrow \mathbb{R}^m$ the constraints function.

The set of points satisfying the constraints is called the feasible region

$$D = \{\mathbf{x} \in \Omega \mid c(\mathbf{x}) \leq 0\}.$$

The problem can be rewritten as

$$\min_{\mathbf{x} \in D} f(\mathbf{x}).$$

Depending on the nature of the objective function and constraints (linear or non-linear, single or multi-objective) of the search space (continuous or discrete), the optimisation problems can be divided in different classes

- **Continuous** or **discrete**: the optimisation variables belong to a feasible set that is a subset of the real space

$$\mathbf{x} \in D \subseteq \mathbb{R}^n.$$

In some problems the variable \mathbf{x} represents integer values. Such problems are defined as **integer programming problems**, and the variables are in a feasible set such that

$$\mathbf{x} \in D \subseteq \mathbb{Z}^n.$$

A subset of integer programming problems is the **binary programming problems** where

$$\mathbf{x} \in D = \{0, 1\}^n.$$

If some of the variables in the problem are not restricted to be integer variables, the problem is called **mixed-integer programming problem**

$$\mathbf{x} = (\mathbf{x}_r, \mathbf{x}_d) \in D \subseteq \mathbb{R}^{n_r} \times \mathbb{Z}^{n_d}, \quad \text{with } n_r + n_d = n.$$

- **Constrained** or **unconstrained**: if there are no constraints on the design variables ($m = 0$), the problem is unconstrained. For constrained optimisation, instead $m > 0$. Unconstrained problems arise also as reformulations of constrained optimisation problems, in which the constraints are added to the objective function as penalisation terms.
- **Linear** or **non-linear**: if the objective function and all the constraints are linear functions of \mathbf{x} , the problem is called **linear programming problem**. Otherwise if some of the constraints or the objectives are non-linear functions, the problem is a **non-linear programming problem**.

- **Global** or **local**: many algorithms for non-linear optimisation problems find only a **local solution**, i.e. a point at which the objective function is smaller than all the other feasible points in a neighbourhood. They do not always find the **global solution**, which is the point that has the lowest function value among all the points of the feasible region. Only for linear programming problems and convex programming problems, the local solution is also the global one. An objective function that presents a large number of local optima is called *multimodal* function.
- **Single-** or **multi-** or **many-**objective: if the objective function is a scalar function, that is

$$n_{\text{obj}} = 1$$

the problem is said to be **single-objective**. In many engineering applications, one is seeking a trade-off between different objectives; f is in this case a vectorial function with

$$n_{\text{obj}} > 1$$

and the problem is called a **multi-objective** optimisation problem ($1 < n_{\text{obj}} \leq 3$) or **many-objective** optimisation problem ($n_{\text{obj}} > 3$). Multi-objective optimisation problems can be transformed into single-objective problems, for example, by means of aggregating functions, condensing all objectives in a single-cost function with the use of weights coefficients, or by using alternatives such as the ϵ -constrained, and the goal-attainment methods. More details are given in Sect. 7.2.3.

To apply the most suitable algorithm, the problem must first be understood and categorised. An algorithm suitable for linear problems may not be suitable for non-linear problems, and vice versa. By incorrectly categorising a problem, an unsuitable optimisation category can be chosen, leading to invalid results, for example, a convex problem. This is a problem where the constraint functions are all convex, all minimising objectives are convex, and all maximising objectives are concave. These problems typically have only one optimal solution, and so every local solution is also a global solution. Using a global algorithm on a convex problem is generally computationally more expensive than a local one while still leading to the correct solution.

When selecting an algorithm, it should be noted also that there is not a single most effective algorithm that can be applied to all optimisation problems. Each algorithm has benefits and drawbacks. The main theorem of optimisation, the *no free lunch theorem* (NFL) [1]), states: if any algorithm A outperforms another algorithm B in the search for an extreme of an objective function, then algorithm B will outperform A over some other desired trait such as computational cost, accuracy or complexity. The NFL theorem suggests that the average performance overall possible objective functions is the same for all search algorithms. All algorithms

for optimisation will give the same average performance when averaged over all possible functions, which means that the universally best method does not exist for all optimisation problems. This theorem proves the importance of applying problem-specific information when deciding upon an appropriate algorithm to achieve better than average results.

7.1.2 *Local vs Global Optimisation*

There are two categories of optimal solutions that can be found as a result of an optimisation process: local solutions and global solutions. Mathematically, a local solution is a solution for which an optimal solution \mathbf{x}_{local}^* is better than all other values of \mathbf{x} in its neighbourhood. A global solution describes an optimal solution \mathbf{x}_{global}^* which is better than all other values of \mathbf{x} across the whole search space. As a result, all global minima are also local minima. This distinction highlights the importance of a correct problem formulation. For a linear, convex problem, a local solution is a global solution. In the case of a complex, non-convex problem, a local solution is not necessarily a global one. In this case the choice of the initial guess, from which the optimisation algorithm performs the search, can be crucial for the performance of the algorithm itself because of the possibility of converging into one of the local optima close to the initial guess rather than the global one. Hence global optimisation algorithms are designed with particular strategies that are aiming at avoiding being trapped in local optima.

7.1.3 *Single- vs Multi-Objective*

The objective functions drive the optimisation algorithm to find an optimum value, depending on whether the result has to be minimised or maximised. In single-objective optimisation, the main goal is to find the ‘optimal’ solution for only one objective function.

For a problem with more than one objective, there is rarely one solution that is the optimal solution for each of the objective functions. In this case, a set of optimal solutions is found. Finding the optimum solution for multiple objective functions can be difficult and computationally expensive. One method of simplification is to reduce the number of objective functions. Multiple objective functions can be lumped into one objective functions through a weighted sum approach, where the function outputs are scaled then multiplied a constant representing its importance relative to the other objectives. It should be noted that, although conceptually easy, the weighted sum approach only finds solution on the convex regions of the Pareto front and are difficult to implement when the objective functions have different orders of magnitude. Another method is the ϵ -constraint one, which considers all objectives except one, as constraints in the optimisation process.

These constraints are assigned different constants based on the importance of their respective objective functions (e.g. minimum reliability levels, maximum price), and multiple solution of a single-objective problem are found for different satisfaction levels of each constraint. A deeper discussion into multi-objective strategies is provided in Sect. 7.2.3.

7.2 Continuous Optimisation

7.2.1 Local Optimisation

Local optimisation algorithms are exact methods that guarantee the convergence to the local optimum in a neighbourhood of search. They are the most investigated optimisation techniques and have their roots in the calculus of variations and the work of Euler and Lagrange. The development of linear programming falls back to the 1940s, and it was the base of the modern optimisation theory that rapidly grew and then was developed in the last 70 years.

As already defined in the previous section, the general optimisation problem is defined as

$$\min_{\mathbf{x} \in D} f(\mathbf{x})$$

where $D = \{\mathbf{x} \in \Omega \mid c(\mathbf{x}) \leq 0\}$, $f : \Omega \rightarrow \mathbb{R}$ and $c : \Omega \rightarrow \mathbb{R}^m$ are sufficiently smooth functions. It must be pointed out that local optimisation techniques restrict their field of application to single-objective optimisation problems with continuous variables. To extend the use to multi-objective optimisation problems, one of the aggregate techniques presented above must be taken into consideration.

Before introducing the optimality results, some definitions need to be stated.

Definition 7.2.1 The real function $\mathcal{L} : \Omega \times \mathbb{R}^m \rightarrow \mathbb{R}$ defined as

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda^T c(\mathbf{x})$$

is the *Lagrangian*, and the coefficients $\lambda \in \mathbb{R}^m$ are called *Lagrange multipliers*.

Definition 7.2.2 Given a point \mathbf{x} in the feasible region, the *active set* $\mathcal{A}(\mathbf{x})$ is defined as

$$\mathcal{A}(\mathbf{x}) = \{i \in \mathcal{I} \mid c_i(\mathbf{x}) = 0\},$$

where $\mathcal{I} = \{1, \dots, m\}$ is the index set of the constraint.

Definition 7.2.3 The linear independence constraint qualification (LICQ) holds if the set of active constraint gradients $\{\nabla c_i(\mathbf{x}), i \in \mathcal{A}(\mathbf{x})\}$ is linearly independent, that is,

$$\text{rank}(\nabla c_i(\mathbf{x}), i \in \mathcal{A}(\mathbf{x})) = |\mathcal{A}|.$$

Note that if this condition holds, none of the active constraint gradients can be zero.

7.2.1.1 Optimality Conditions

These definitions allow the statement of the following optimality conditions (refer to [2], for a proof of the Theorems).

Theorem 7.2.1 (First-order necessary condition) *Suppose that \mathbf{x}^* is a local solution of the constrained non-linear programming (NLP) problem and that the LICQ holds at \mathbf{x}^* . Then a Lagrange multiplier vector λ^* exists such that the following conditions are satisfied at the point $(\mathbf{x}^*, \lambda^*)$*

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*) = 0, \quad (7.1)$$

$$c(\mathbf{x}^*) \leq 0, \quad (7.2)$$

$$\lambda^* \geq 0, \quad (7.3)$$

$$(\lambda^*)^T c(\mathbf{x}^*) = 0. \quad (7.4)$$

These conditions are known as the Karush-Kuhn-Tucker (KKT) conditions.

Remark 7.2.1 The last condition implies that the Lagrange multipliers corresponding to inactive inequality constraints are zero; hence it is possible to rewrite the first equation as

$$0 = \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) - \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i^* \nabla c_i(\mathbf{x}^*).$$

The optimality condition presented above gives information on how the derivatives of objective and constraints are related at the minimum point \mathbf{x}^* . Another fundamental first-order necessary condition that gives additional information on the gradient of the objective function in the optimal point can be stated. For this an additional definition is needed.

Definition 7.2.4 Given a feasible point $\mathbf{x} \in D$, a sequence $\{\mathbf{x}_k\}_{k=0}^{\infty}$ with $\mathbf{x}_k \in \Omega$ is a *feasible sequence* if, for all $k \in \mathbb{N}$, $\mathbf{x}_k \in D \setminus \{\mathbf{x}^*\}$ and

$$\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}.$$

Given a feasible sequence, the set of the limiting directions $w \in \Omega \setminus \{0\}$

$$\lim_{k \rightarrow \infty} \frac{\mathbf{x}_k - \mathbf{x}}{\|\mathbf{x}_k - \mathbf{x}\|_2} = \frac{w}{\|w\|_2}$$

is called the *cone of the feasible directions*, $C(\mathbf{x})$.

Moving along any vector of this cone (with vertex in a local minimum point \mathbf{x}^*) either increases the objective value or keeps it the same.

Theorem 7.2.2 (First-order necessary condition) *If \mathbf{x}^* is a local solution of the optimisation problem and f is differentiable in \mathbf{x}^* , then*

$$\nabla f(\mathbf{x}^*) \cdot w \geq 0 \quad \forall w \in C(\mathbf{x}^*).$$

For the directions w for which $\nabla f(\mathbf{x}^*) \cdot w = 0$, it is not possible to determine, from first derivative information alone, whether a move along this direction will increase or decrease the objective function. It is necessary to examine the second derivatives of the objective function and constraints to see whether this extra information resolves the issue. The directions for which the behaviour of f is not clear from the first derivative form the following set:

Definition 7.2.5 Given a pair $(\mathbf{x}^*, \lambda^*)$ satisfying the KKT conditions

$$C(\lambda^*) = \{w \in C(\mathbf{x}^*) \mid \nabla c_i(\mathbf{x}^*) \cdot w = 0, \text{ for all } i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}, \text{ with } \lambda_i^* > 0\}$$

is called the *critical cone*.

Indeed for $w \in C(\lambda^*)$ from the first KKT condition it follows that

$$\begin{aligned} \nabla f(\mathbf{x}^*) \cdot w &= \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i^* \nabla c_i(\mathbf{x}^*) \cdot w \\ &= 0. \end{aligned}$$

If \mathbf{x}^* is a local solution, then the curvature of the Lagrangian along the directions in $C(\lambda^*)$ must be non-negative in the case of qualified constraints. A positive curvature is instead a sufficient condition for a local optimum.

Theorem 7.2.3 (Second-order necessary condition) *Let f and c be twice continuously differentiable; \mathbf{x}^* is a local solution of the constrained problem and that the LICQ condition is satisfied. Let $\lambda^* \in \mathbb{R}^m$ be the Lagrange multiplier for which the pair $(\mathbf{x}^*, \lambda^*)$ satisfies the KKT conditions. Then*

$$w^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}^*, \lambda^*) w \geq 0, \quad \forall w \in C(\lambda^*)$$

Theorem 7.2.4 (Second-order sufficient condition) *Let f and c be twice continuously differentiable; \mathbf{x}^* is a feasible point, $\lambda^* \in \mathbb{R}^m$ such that $(\mathbf{x}^*, \lambda^*)$ satisfies the KKT conditions and*

$$w^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}^*, \lambda^*) w > 0, \quad \forall w \in C(\lambda^*), w \neq 0.$$

Then \mathbf{x}^ is a strict local minimum of the constrained problem.*

7.2.1.2 Algorithms

In the last 50 years, a variety of approaches have been developed to solve NLP problems, first tackling the most simple unconstrained NLP problem and then expanding their application also to the constrained case. A starting point, denoted by \mathbf{x}_0 , is always provided to the algorithm by the knowledge of the user or left to the optimiser. The optimisation process iterates exploiting information on the objective, constraints, their derivatives and the previous iterates to terminate whenever no further progress can be made or the optimal solution is approximated with acceptable accuracy.

The algorithm for unconstrained NLP is presented first. They are divided into two groups: line search based and trust region.

- **Line search:** the algorithm determines a search direction p_k and searches along this direction from the current iterate \mathbf{x}_k for a new iterate with a lower function value. The step length to move along p_k can be found by approximately solving the minimisation problem

$$\min_{\alpha > 0} f(\mathbf{x}_k + \alpha p_k).$$

At the new point, a new search direction and step length are computed, and the process is repeated until convergence.

- **Trust region:** the algorithm constructs a model function m_k whose behaviour near the current iterate x_k is similar to that of the actual objective function f . The iteration direction of search p is found as the solution of the problem

$$\min_{p \in \Omega} m_k(\mathbf{x}_k + p),$$

where $\mathbf{x}_k + p$ lies inside the trust region. If the solution does not produce a sufficient decrease in f , it means that the trust region is too large. In this case the trust region is shrunk and the minimisation problem is solved again. Usually the trust region is the ball

$$\|p\|_2 \leq \Delta, \quad \text{where } \Delta \text{ is the trust region radius}$$

and the model m_k is usually a quadratic function of the form

$$m_k(\mathbf{x}_k + p) = f(\mathbf{x}_k) + p^T \nabla f(\mathbf{x}_k) + \frac{1}{2} p^T H(\mathbf{x}_k) p$$

where H is the Hessian matrix of the Lagrangian.

The two approaches differ in the way they choose the direction and the distance of the move: line search based fixes the direction p_k and optimises the length of the step. Thrust region instead first chooses the maximum distance of the move, the trust

region radius, and then seeks for the best move to attain the best improvement of the objective function.

As an example for line search methods, there are (refer to [2] for the details about the methods):

- **Steepest descent method:** it chooses as search direction the descent one $p_k^{SD} = -\nabla f(\mathbf{x}_k)$.
- **Newton methods:** the search direction is the solution of the Newton equation $p_k^N = -H(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$.
- **Non-linear conjugate gradient methods:** where the search direction is defined as $p_k^{CG} = -\nabla f(\mathbf{x}_k) + \beta_k p_{k-1}$ with $\beta_k \in \mathbb{R}$.
- **Quasi-Newton methods:** they don't require the computation of the second-order derivatives but use an approximation of it (B), $p_k^{QN} = -B(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$; quasi-Newton methods significantly increase convergence speed compared with Newton ones.

Newton and quasi-Newton methods are the ones that attain a superlinear rate of convergence, but they require the computation (or approximation) and the storage of the Hessian matrix. On the other hand, the methods that rely just on the gradient information are slower at convergence.

Most of the methods have a counterpart for the thrust region approach. In the quadratic model, the Hessian matrix is substituted by the one used by each method (identity matrix for the steepest descent, H_k for the Newton method and its approximation B_k for quasi-Newton methods). It is possible to prove that the resulting search direction is defined as in the line search methods and its length constrained by the trust region radius.

The presentation of the algorithms for the unconstrained case was necessary to introduce the techniques for solving constrained NLP problems as parts of them rely on the idea of converging to the solution of the constrained problem by approximating it with a sequence of unconstrained problems.

The algorithms for constrained NLP problems can be grouped in:

- **Penalty, barrier, augmented Lagrangian methods and sequential linearly constrained methods:** they solve a sequence of simpler subproblems (unconstrained or with simple linearised constraints) related to the original one. The solutions of the subproblems converge to the solution of the primal one either in a finite number of steps or at the limit.
- **Newton-like methods:** they try to find a point satisfying the necessary conditions of optimality (KKT conditions in general). The sequential quadratic programming (SQP) method is part of this class.

The *penalty methods* combine the objective function and constraints into a penalty function $\alpha(\mathbf{x})$ which is null for feasible points and positive otherwise. The problem to be minimised is the unconstrained problem

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) + \mu\alpha(\mathbf{x})$$

for a series of increasing values of the penalty parameter μ , such that $\mu\alpha(\mathbf{x}) \rightarrow 0$ as $\mu \rightarrow \infty$, until the solution of the constrained optimisation problem is identified with sufficient accuracy. From a computational point of view, superlinear convergence rates might be achieved, in principle, by applying Newton's method to solve the minimisation problem (or its variants such as quasi-Newton methods). The algorithmic behaviour is strictly related to the choice of the penalty parameter. If μ is large, more importance is given to the feasibility than the optimality, and the iterates could move to feasible regions far from the optimum, causing slow convergence and premature termination.

The *barrier methods* or *interior-point methods* add terms to the objective function that act as a barrier and prevent the iterates from leaving the feasible region. For example, in the case of inequality constrained problems, a barrier problem can be formulated as

$$\min_{\mathbf{x} \in \Omega} \theta(\mu),$$

where $\mu \geq 0$ and $\theta(\mu) = \inf\{f(\mathbf{x}) + \mu b(\mathbf{x}) : c_i(\mathbf{x}) < 0, \forall i \in \mathcal{I}\}$. The barrier function b should be non-negative and continuous on the feasible region and go to infinity as the boundary is approached from the interior. This would guarantee that the iterates do not leave the domain. The starting point must be chosen in the interior of the feasible region, and the Newton or quasi-Newton methods can solve the successive barrier problem.

In the *augmented Lagrangian methods*, a penalty functions is added to the Lagrangian:

$$\mathcal{L}_A(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) - \lambda^T c(\mathbf{x}) + \frac{1}{2\mu} \|c(\mathbf{x})\|_2^2$$

Fixing λ to some estimate of the optimal Lagrange multipliers and $\mu > 0$ to some positive value, it is possible to find a value of \mathbf{x} that approximately minimises $\mathcal{L}_A(\cdot, \lambda, \mu)$. Then the process is repeated updating λ and μ with the information from the previous \mathbf{x} -iterate.

In *sequential linearly constrained methods*, at every iteration, a Lagrangian is minimised subject to a linearisation of the constraints.

The *sequential quadratic programming* has instead a completely different approach. It employs Newton-like methods to solve directly the KKT conditions of the original problem. The problem turns out to be a minimisation problem of a quadratic approximation of the Lagrangian subject to a linear approximation of the constraints. The search direction p_k at the iterate $(\mathbf{x}_k, \lambda_k)$ is the solution of the problem

$$\begin{aligned} \min_p \quad & \frac{1}{2} p^T \nabla_{xx}^2 \mathcal{L}(\mathbf{x}_k, \lambda_k) p + \nabla f(\mathbf{x}_k) \cdot p \\ \text{s.t.} \quad & \nabla c_i(\mathbf{x}_k) \cdot p + c_i(\mathbf{x}_k) \leq 0, \quad i \in \mathcal{I}. \end{aligned}$$

A trust region constraint can be added to the algorithm to control the length of the step, and a quasi-Newton approximation of the Hessian can be used instead of the second derivatives of the Lagrangian.

7.2.2 *Global Optimisation*

The effectiveness of the traditional local optimisation techniques on multimodal objective functions strongly depends on the initial guess solution given to a method. If a previous knowledge of the problem is available, the designer can provide a good initial guess to the algorithm to ensure convergence to the global optimum. Otherwise the algorithm will mostly fail in the global search, getting trapped in one of the multiple local minima.

The purpose of global optimisation is to find the best solution of a non-linear optimisation problem,

$$\min_{\mathbf{x} \in D} f(\mathbf{x})$$

in the presence of multiple optima and a non-smooth objective function.

Nevertheless, local optimisation techniques will often play an important role also in global optimisation strategies since some promising global approaches combine both global and local strategies of search. This is the case, for example, for memetic algorithms (MA) [3] that combine gradient-based technique with evolutionary algorithms: the global search generates a set of trial points over the feasible region (solutions of the evolutionary strategy), and the local algorithm performs local descent search from the best available points, in an iterative loop that alternates the two steps till convergence. Obviously the best compromise between global and local strategies and the effectiveness of their use depends on the characteristic of the problem such as the geometry of the feasible region, the number of local minima and the sharpness of the objective function in the neighbourhood of the global solution. However, the collaborative use of the global exploration capabilities of the first algorithm to prune the search space narrowing the area of search and the exploitation of local strategies to converge to the exact location of the global minimum is a very simple but effective approach for the local refinement of the selected global optimal solutions.

The limit of combining the two optimisation approaches may be related to the inefficiency of the local strategies in dealing with multiple-objective problems, especially when proper scalarisations are not considered. First, a brief overview to the available global optimisation algorithms and to the historical background that made them evolving to the actual state of the arts is given (see [4, 5] for a complete survey).

The methods that were first used in global optimisation were deterministic techniques. They were introduced in the late 1950s with the advent of the first

electronic computers into the research community. They are mostly based on the idea of trying to construct a sequence of approximate solutions which converge to the exact one by dividing the problem into smaller subproblems or approximations of the original one. With the evolution of the computational power at the beginning of the 1990s, different probabilistic global optimisation approaches were affirmed as new strategies. Among them it is worth to mention simulated and nested annealing [6, 7] and the large family of evolutionary strategies [8]. They are in general computationally less efficient than deterministic techniques, but due to their structure, they are able to tackle a wider range of problems, and no assumptions on the model regularity and smoothness is required. For these reasons they are considered as one of the most promising techniques for solving also global discrete non-linear optimisation problems.

There are several classifications of global optimisation strategies. One is the already mentioned division between **deterministic** and **stochastic** algorithms. In the first category, the model and the optimisation variables are completely known, and the algorithm performs through predefined steps. The stochastic component of the latter group instead lies either on the random sampling of the trial points, random parameters of the algorithm itself that made the single step not predictable, or on the use of a stochastic model for the objective function. Another division can be made between **exact** methods and **heuristic** methods. Exact methods provide a mathematical proof that the optimal solution can be found, while heuristic methods are not based on convergence theories. In most of the cases, no guarantee of finding the optimal solution can be provided and used to stop the search process: the optimisation process is constituted of iterative steps that improve the candidate solutions based on a measure of the quality of their fitness, a function that combines indexes of optimality and feasibility.

An overview of the relevant methods is given below according to the first classification deterministic or stochastic with an internal differentiation between exact and heuristic methods. The objective of the section is to give a comprehensive overview of the available methodologies. For details about a specific algorithm, please refer to the corresponding bibliography. The extension of the methodologies to the multi-objective case is discussed in the next section.

7.2.2.1 Deterministic Strategies

The first group are deterministic and exact global optimisation strategies [9]. It means that no randomness is involved in the optimisation process and the algorithm will always produce the same solutions for the same starting condition or initial state. The optimisation steps are predictable and a proof of convergence exists.

- **Uniform grid search** [4]: it is a trivial search strategy that makes use of a grid over the search domain to evaluate cost and constraints functions. Local search from a point in each element of the grid can be performed, and the feasible local minimum with lowest objective function is the approximation of the global

optimum. The success of the local search obviously depends on the finesse of the search grid, and global convergence can be trivially guaranteed by the fact that the mesh can be made arbitrarily dense. Such a simple scheme however rapidly becomes inefficient with the enlargement of the bounds on the optimisation variables and the raising of the dimension. The computational load will increase as an exponential function of the dimensionality of the problem.

- **Complete (enumerative) search** [5]: it is based on the simple principle of searching through all potentially optimum points in the search space, through enumeration of the possible candidates and evaluation of the objective. If, for example, the feasible region D is a polyhedra and the objective function is concave, then it is possible to prove that the problem must have a global optimal solution which is a corner of D . Since D has a finite number of extreme points, the problem could be solved by enumerating the extreme points of D in an appropriate way until an optimal solution is found [10]. Enumerative methods have few applications in continuous optimisation. Convergence properties are trivially provable.
- **Homotopy and trajectory methods** [11, 12]: the two strategies have the ambitious objective of visiting all stationary points of the objective function on the feasible domain, tracing the paths on the feasible space that include them. The solutions are then explored through enumeration techniques and evaluation of the objective. The two methods differ in the way of constructing their paths: the homotopy method makes use of homotopy transformations between the solution of a simplified problem and the original one; the trajectory problem solves a set of ordinary differential equations. The methodologies are applicable to smooth problems with continuous variables, and the enumeration techniques employed guarantee convergence to the optimum.
- **Sequential approximation (relaxation) methods** [13]: the idea is to build and solve a series of approximate (or relaxed) optimisation subproblems converging to the exact (or approximate) global optimum. A classification of such methods is based on the target of the approximation (relaxation), either specific model parameters or the entire system and subsystem models in a non-decomposed or decomposed problem, and the method employed to perform the approximated model fitting (response surface methodology (RSM), Taguchi methods, kriging [14]). The methods can be applied to a wide range of optimisation problems with continuous and discrete variables, and they are particularly suitable for expensive or noisy simulation models as a complete analysis is performed only in the experimental data points of the metamodeling techniques. The methods form a subset of the derivative-free optimisation techniques, based on model approximation, as they are completely free from derivative computation or approximation. Method-specific convergence theories are available in the suggested reference.
- **Interval arithmetic methods** [15]: it is possible to develop a complete theory based on interval entities analogous to the real one. The strength of exploiting the global information over large domains given by interval analysis in optimisation methods ensures the convergence to all global optima. The idea is to start with

an initial box and to delete the sub-boxes that cannot contain the global solution by a branch-and-bound procedure. The process terminates, when the bounds on the solutions and on the global minimum are below a predefined tolerance. The main drawback of the interval approach is its computational complexity. It is applicable to MINLP problems and non-smooth functions.

On the other hand, there is no proof of exactness for the following global deterministic strategy.

- **Sequential improvements of local optima** [16]: the basic idea is to generate an improving sequence of local minima. Deflection techniques, tunnelling and filled function methods are examples of this approach. The tunnelling method consists of two phases: seek for a local minimum and apply a tunnelling function to find a point in the domain that has the same value of the objective function. The newly formed point is the starting point for the next iteration. The process terminates when it is not possible to detect any point during the second phase. The last found local optimum is also the global one. There is no rigorously established convergence theory associated with these methods, and they are applicable only to smooth continuous optimisation problems.

7.2.2.2 Stochastic Strategies

Stochastic strategies are methods that contain not deterministic elements, either random generated algorithm parameters or stochastic approximations of model functions. As expected it is difficult to develop a rigorous convergence theory for such a class of algorithms, due to the randomness introduced in the optimisation process. However two of them provide a convergence proof based on probabilistic theories, and they can be classified as exact methods.

- **Random search methods** [17]: the objective of these search methods is to find the global minimum with an adaptive-probabilistic distribution of random points over the feasible region. These algorithms ensure that the global minimum will be found with probability one as the sample size grows to infinity. The difference to the deterministic grid search algorithm lies in its adaptivity. The number of experimental points doesn't need to be decided in advance, but it is generated in the successive steps. These methods are applicable to both discrete and continuous global optimisation problems with very mild assumptions on the model regularity.
- **Random function approach** [18, 19]: also known in literature as Bayesian methods, they are the stochastic counterpart of the sequential approximation approach with an adaptive probabilistic model for the approximation of the objective function. They are suitable for cost functions that have a highly computational load. They can deal with continuous and discrete variables and non-smooth functions. A theoretical convergence to the global optimum is guaranteed only by generating a dense set of search points.

The larger group of stochastic optimisation techniques are heuristic. They are most widely applied in practice, but in general no mathematical proof of convergence exists. However, some results on convergence for evolutionary methods, provided that the method satisfies some very general conditions, have been published for single- [20] and multi-objective [21] problems.

- **Two-phase methods** [22]: they are the stochastic counterpart of the deterministic grid search technique. They combine two phases of search: a global one and a local one. The process starts with a random sampling of the feasible space followed by the application of a local refinement. Multistart [22], clustering methods [23] and multilevel single linkage [24] are the examples. The range of applications for the technique is constrained to the local search used. The greedy global strategy is suitable for both continuous and discrete variables with no assumptions on the model structure.
- **Simulated annealing** [25]: the technique is based on the analogy between minimising a cost function and the cooling process of a material till it reaches its state of low energy equilibrium. The algorithm iteratively brings the actual state (optimisation variables) to a lower level of the internal energy of the system (objective function). The changes between the states are done probabilistically. The new configuration is constructed by imposing a random displacement at each step. If the energy of the new state is lower than the previous one, the change is accepted. If the energy is greater, the new configuration is accepted with a probabilistic value. The probabilistic acceptance of upward moves is aiming to avoid the convergence to the local minima. It is able to tackle global optimisation problems with discrete and continuous variables under mild assumptions on the model regularity.
- **Genetic algorithms (GAs)** [26]: are stochastic search methods that take their inspiration from natural selection and survival of the fittest in the biological world. Each iteration of a GA involves a competitive selection that eliminates poor solutions. The solutions with high fitness are recombined with other solutions by swapping parts of a solution with another. The solutions are also mutated by making a small change to a single element, or a small number of elements, of the solution. Recombination and mutation are used to generate new solutions that are biased towards the regions of the space for which good solutions have already been seen. GAs were born and are well suited, to solve discrete problems, and they have been successfully applied to continuous problems as well. Most of their efficacy is due to a powerful recombination operator, which, for this reason, becomes the main operator. The recombination operation used by GAs requires that the problem can be represented in a manner that makes combinations of the two solutions likely to generate interesting solutions. Selecting an appropriate representation is a challenging aspect to properly apply these methods. Usually a binary coding is used, and many applications have demonstrated the validity of this approach.
- **Estimation of distribution algorithms (EDA)**: with the idea that probabilistic modelling may offer a more efficient/effective way to treat real problems,

instead of using standard genetic operators used in traditional EAs, in EDAs new candidate solutions to the problem are generated using *regression*, i.e. estimating a probabilistic model based on the statistics collected from the set of candidate solutions (regression), and *sampling* the achieved probabilistic model, bringing a new paradigm in evolutionary computation. Because of the different natures of both optimisation and probabilistic modelling in discrete and continuous domains, developed EDAs also have differences depending on the representation type they use for the problem. Many of the early continuous EDAs as well as their recent improvements are based on the assumption that design variables can be characterised by Gaussian distribution. The continuous population-based incremental learning (PBIL_C) [27] extends the original discrete version to continuous domains by updating a vector of independent Gaussian distributions. The continuous univariate marginal distribution algorithm (UMDA_C) [28] uses maximum likelihood estimation to learn the parameters of the Gaussian distribution for each variable from the population of solutions. The continuous mutual information maximisation for input clustering (MIMIC_C) [28] learns the chain structured probabilistic model for continuous variables by adapting the concept of conditional entropy for univariate and bivariate Gaussian distributions.

Other probabilistic models estimate a non-parametric distribution for the variables have also been used in continuous EDAs. The multi-objective Parzen-based estimation of distribution (MOPED) [29] uses a Parzen estimator to build the probabilistic model. Both Gaussian and Cauchy kernels are used alternatively during evolution to exploit their complementary characteristics.

A review of methods and their characteristics can be found at [30].

- **Differential evolution (DE)** [31]: it is an optimisation method particularly suitable for multidimensional multimodal functions, belonging to the class of evolution strategy (ES). The main idea is to generate a variation vector by taking the weighted difference between two other solution vectors randomly chosen within a population of solution vectors and to add that difference to the vector difference between the considered solution and a third solution vector.

An approach used to create new algorithms is to hybridise existing ones by appropriately mixing some of their building blocks. By following this approach, and based on some new theoretical results on the convergence of DE, the inflationary differential evolution algorithm (IDEA) [32] was proposed, combining DE with the restarting procedure of monotonic basin hopping (MBH) algorithm [33, 34]. Although IDEA showed very good results when applied to problems with a single or multi-funnel landscape, its performance was found to depend on the parameters controlling both the convergence of DE and MBH and the inflationary stopping criterion used to terminate the DE search.

Despite its simplicity, the standard DE alone shows good performance on a broad range of problems featuring multimodal, separable and non-separable structures, but the performance is strongly influenced by three parameters: the population size, n_{pop} ; the crossover probability, CR ; and the differential weight (or step parameter), F . In addition, it was reckoned that the chosen strategies for mutation and crossover [35] plays an important role.

The need of self-adapting techniques especially for these two parameters has been widely recognised in the literature. In [36] the authors introduced a fuzzy adaptive differential evolution algorithm using fuzzy logic controllers to adapt the parameters for the mutation and crossover operators. The self-adaptive DE (SADE), described in [37], incorporates a mechanism that self-adapts both the parameters CR and F and the trial vector generation strategy. In [38] an adaptation strategy is proposed for parameter F , while CR is kept constant. In [39] both control parameters are added to each individual of the population and evolve with it. An alternative approach for the on-line adaptation of both CR and F parameters and embedded into the general framework of IDEA is proposed in [40]. The proposed approach uses the Parzen kernel method to build a joint probabilistic representation of the most promising region of the bivariate $CR - F$ space. The resulting probability density function (PDF) is updated during the optimisation process on the basis of obtained results. A further development of AIDEA is multi-population adaptive inflationary differential evolution algorithm (MP-AIDEA) [41] where multiple populations are initialised in the search space and exchange information during the optimisation process.

- **Particle swarm optimisation (PSO)** [42]: it is a population-based stochastic optimisation technique developed by Eberhart and Kennedy in 1995 [43], inspired by the social behaviour of bird flocking or fish schooling. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution it has achieved so far. The particle swarm optimisation concept consists of, at each iteration, changing the velocity of each particle i according to a close-loop control mechanism.

7.2.3 Multi-Objective Optimisation

The problem of optimising concurrently two or more objective functions falls into the category of multi-objective optimisation problems. In contrary to single-objective optimisation, the purpose is not to find a unique global optimal solution but rather a set of solutions representing the compromise (trade-offs) between the different objectives.

Also in multi-objective optimisation, as in single-objective, it is possible to distinguish between local and global solutions: they will be referred as global frontier and local frontier.

The generic multi-objective optimisation problem is defined as

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0 \end{aligned}$$

where \mathbf{x} is the optimisation variables vector, $f : \Omega \rightarrow \mathbb{R}^{n_{\text{obj}}}$, with $n_{\text{obj}} > 1$, the objective function and $c : \Omega \rightarrow \mathbb{R}^m$ the constraints function. As before the set of feasible points is denoted by D .

To extend the methodologies presented for global optimisation to the multi-objective case, it is necessary to introduce some definitions.

Definition 7.2.6 A point $\mathbf{x}_1 \in D$ *Pareto dominates* $\mathbf{x}_2 \in D$ if

$$f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2), \quad i = 1, \dots, n_{\text{obj}}$$

and there is at least one component $j \in \{1, \dots, n_{\text{obj}}\}$ such that

$$f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2).$$

This is indicated by

$$\mathbf{x}_1 \preceq \mathbf{x}_2.$$

Definition 7.2.7 A point $\mathbf{x}^* \in D$ is *Pareto optimal* if it isn't dominated by any $\mathbf{x} \in D$,

$$\mathbf{x} \preceq \mathbf{x}^*.$$

In other words, a solution is said to be Pareto optimal, or equivalently *nondominated*, if there is no other point in the feasible space for which a decrease in one objective will not cause a simultaneous increase of at least one of the other objectives.

Definition 7.2.8 For a multiple objective optimisation problem, the *Pareto optimal set* is defined as

$$\mathcal{P}^* = \{\mathbf{x} \in D \mid \nexists \mathbf{x}' \in D \ \mathbf{x}' \preceq \mathbf{x}\}.$$

Definition 7.2.9 The union of the objective values of all Pareto optimal points is called *Pareto front* or equivalently

$$\mathcal{P}\mathcal{F}^* = \{f(\mathbf{x}) \in \mathbb{R}^{n_{\text{obj}}} \mid \mathbf{x} \in \mathcal{P}^*\}.$$

The Pareto front is the set of all solutions in the feasible space that are not dominated by any other possible solution. The minima in the sense of Pareto will lie on the boundary of the feasible region or in the tangent points of the objective functions. Generally it is not possible to derive analytically the equation of the front. Approximation techniques have been developed during the years to approach the Pareto frontier by successive iterations or to solve in parallel a sequence of single-objective optimisation problems.

A comprehensive survey of multi-objective optimisation techniques is given in [44–46], the last two focusing mainly on global evolutionary multi-objective

strategies. Evolutionary programming is the area of multi-objective optimisation research that in the last years registered the fastest growth. This is due to the intrinsic structure of the evolutionary algorithms, population based, well suited for an extension to multi-objective problems.

The multi-objective approaches are divided in methods that use the concept of Pareto dominance for the selection mechanism of the next iterates and methods that develop a special handling of the objective functions for reformulating the problem as single objective. The latter techniques are applicable to all the presented global optimisation strategies, while the former are typically for evolutionary algorithms.

The aggregation of the multiple objectives into a common single objective can be achieved by the different techniques presented below, outlining their main advantages and disadvantages.

- **Weighted sum approach:** the objectives are aggregated into a single function using weighting coefficients. The optimisation problem becomes

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & \sum_{i=1}^{n_{\text{obj}}} w_i f_i(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0, \end{aligned}$$

where $w_i \geq 0$ and it is usually assumed that

$$\sum_{i=1}^{n_{\text{obj}}} w_i = 1.$$

By varying the values of the coefficients, different solutions on the Pareto front are traced. To cover the entire front, a sequence of single-objective optimisation problems needs to be solved, making the procedure very inefficient from a computational point of view. Moreover, this technique has the drawback of not generating proper Pareto optimal solutions in the presence of non-convex search spaces [47]. Additionally, there is no a priori knowledge about how a change in the weights will affect the position on the Pareto front of the new solution.

- **Goal programming [48]:** the designer has to assign targets to the objectives, and the optimisation problem is transformed in the problem of minimising the sum of the norms of the deviations from the targets

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & \sum_{i=1}^{n_{\text{obj}}} \|f_i(\mathbf{x}) - T_i\|_2 \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0. \end{aligned}$$

Prerequisite in the application of such a technique is a deep knowledge about the optimisation problem to be able to assign meaningful target values to the objectives. The search space is explored by varying the T_i targets, and convergence to the Pareto front is achieved with a prior knowledge of the problem, to assign the targets close to the objectives values of the Pareto optimal points.

- **Goal attainment:** it is a combination of the previous two techniques. Objectives goals are assigned as before, together with relative under or over attainment weight coefficients. The problem becomes

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & \alpha \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0 \\ & f_i(\mathbf{x}) \leq T_i + \alpha w_i, \quad i = 1, \dots, n_{\text{obj}}, \end{aligned}$$

where $\alpha \in \mathbb{R}$ and the weights $w_i \geq 0$ are normalised so that

$$\sum_{i=1}^{n_{\text{obj}}} w_i = 1.$$

It is possible to prove that the Pareto front can be covered varying the weight coefficients and the methodology is able to deal also with non-convex problems [49].

- **The ε constraint method:** the objectives are minimised one at a time, constraining the others below a certain level

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & f_j(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0 \\ & f_i(\mathbf{x}) \leq \varepsilon, \quad i = 1, \dots, n_{\text{obj}}, \quad i \neq j. \end{aligned}$$

The main weaknesses of the approach are the same as listed above, computational efficiency, and a necessary a priori knowledge of the problem for covering the global Pareto front.

- **Lexicographic order:** the objectives are sorted by user intervention. The optimisation problem is divided in n_{obj} subproblems solved sequentially with a pre-established order and with additional constraints for not violating the satisfaction of the minimum values of the former subproblems. Assuming that $\{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_{n_{\text{obj}}}(\mathbf{x})\}$ are the ordered objectives and f_i^* the minimum value achieved for the i -th objective. Then the i -th subproblem is defined as

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \quad & f_i(\mathbf{x}) \\ \text{subject to} \quad & c(\mathbf{x}) \leq 0 \\ & f_j(\mathbf{x}) = f_j^*, \quad j = 1, \dots, i - 1. \end{aligned}$$

To cover the Pareto front, different optimisation runs with different sequences of objectives must be performed, heavily increasing the overall computational time.

- **Game theory:** a ‘player’ is assigned to each objective function. The player has the goal to minimise its objective. Assuming that the players are playing a non-cooperative game (i.e. the players make decisions independently), the

intersection of the best strategy of each player is a *Nash equilibrium*, in the sense that no player can deviate unilaterally from this point for further improvement of the proper objective.

- **Weighted min-max approach:** the deviations from the attained minima, in the n_{obj} single-objective subproblems are estimated for the i -th objective as

$$\bar{z}_i(\mathbf{x}) = \frac{\|f_i(\mathbf{x}) - f_i^*\|_2}{\|f_i^*\|_2}, \quad \bar{\bar{z}}_i(\mathbf{x}) = \frac{\|f_i(\mathbf{x}) - f_i^*\|_2}{\|f_i(\mathbf{x})\|_2}$$

assuming that the objective values do not vanish.

Defining $z_i(\mathbf{x}) = \max\{\bar{z}_i(\mathbf{x}), \bar{\bar{z}}_i(\mathbf{x})\}$, the desirable solution of the multi-objective problem is the one that gives the smallest values of all increments of all the objective functions

$$\min_{\mathbf{x} \in D} \max_{i \in \mathcal{I}} \{z_i(\mathbf{x})\},$$

where \mathcal{I} is the set of the objective indexes. The entire front can be covered by weighting the deviation function.

Note that some of the scalarisation approaches, such as the weighted sum, the goal attainment and the ϵ constraint, can be obtained as particular cases of the Pascoletti–Serafini scalarisation scheme [50, 51].

The exploitation of the concept of Pareto dominance in the population-based strategies led in the current years to the development of efficient multi-objective global optimisation techniques. The particular structure of the algorithms, based on a family of solutions that evolves at each step, made the introduction of the concept of Pareto dominance in its ranking process possible [52]. The basic idea is to find a set of solutions that are Pareto nondominated by the rest of the solutions of the feasible set, assign to them the highest rank and remove them from the group. The process then repeats recursively for lower values of the rank. This procedure can be applied for sorting the solutions of a current iteration and selecting a subgroup from it to apply the criteria of evolution of the species, resulting in a next generation of solutions that is different from the previous one and has an average better fitness.

Genetic algorithms are the larger class of evolutionary algorithms. They are divided in two groups:

- **First generation:** they are characterised by the introduction of the concept of Pareto dominance in the process of selection of the population and for the niching operator to maintain the diversity and avoid premature convergence to local fronts. Representative algorithms of this class are multi-objective genetic algorithm (MOGA) [53], nondominated sorting genetic algorithm (NSGA) [54] and niched Pareto genetic algorithm (NPGA) [55].
- **Second generation:** they exploit the concept of *elitism*. This means that they use an external archive to store the nondominated solutions found in the previous generation in a way that the best solutions found in every iteration cannot be lost

during successive iterations and a better global minima frontier can be achieved. The algorithms then differ in the way they interact with the external population. Representative algorithms of this class are strength Pareto evolutionary algorithm (SPEA) [56, 57], NSGA2 [58], Pareto archived evolution strategy (PAES) [59], Pareto envelope-based selection algorithm (PESA) [60, 61] and micro-genetic algorithm (Micro-GA) [62, 63].

Another group of population-based algorithms not classifiable as genetic algorithms already mentioned in the previous section gets inspired by natural phenomena such as the cooling state of a metal or the behaviour of an ant colony in the search of food. A corresponding reformulation of the already presented algorithms is available for multi-objective optimisation problems. Namely, they are multi-objective simulating annealing (MOSA) [64], multi-objective particle swarm optimisation (MOPSO) [65] and multi-objective ant colony optimisation (MOACO) [66].

7.2.4 Optimal Control

The general statement of an *optimal control problem* (OCP) requires the definition of [67]:

- The mathematical model of the dynamic system to control
Usually it is described by a system of ordinary differential equations (ODEs) in the form $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$. The independent variable has been indicated by t , usually appointed as time, but there is no restriction on its choice. The variables x_i in the vector of \mathbf{x} are usually called *state variables*, while u_j in the vector \mathbf{u} are the *control variables*.
- The performance index J to be minimised (or equivalently maximised)
The performance index in the general form is written as:

$$J = \phi[t_f, \mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)]dt \quad (7.5)$$

The *optimal control problem* is in the Bolza form if both the end-cost and the integral terms are present. If the end-cost term ϕ is zero, it is known as a Lagrange problem. On the contrary, if the integral term L is zero, the problem is referred as a Mayer one. Mathematically these formulations are equivalent and convertible into each other. For example, a Lagrange problem can be restated as a Mayer one by simply adding one state variable of the form $\dot{x}_{n+1} = L[t, \mathbf{x}(t), \mathbf{u}(t)]$, leading to $J = x_{n+1}(t_f)$. However, [68] states that, even if they are mathematically equivalent, they are not numerically corresponding. The Lagrange form shall be preferred as the Mayer form leads to an increased number of state variables, which are then discretised in numerical methods, leading to a higher size of the NLP subproblem and a more time-consuming algorithm.

- Specification of constraints

They are divided into two different classes, i.e. *fixed-event* or *path* constraints. The first type is described as an algebraic function of the state and control $g_L^f \leq g^f[(\bar{t}_j), \mathbf{y}(\bar{t}_j), \mathbf{u}(\bar{t}_j)] \leq g_U^f$ at a fixed time \bar{t}_j . The initial and final boundary conditions fall into this form for $g_L^f = g_U^f$. A *path constraint* is formulated as an algebraic function of the state and control variables $g_L^p \leq g^p[t(t), \mathbf{x}(t), \mathbf{u}(t)] \leq g_U^p$ over a trajectory's phase. Bounds on the control magnitude fall into this category as $\mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U$. This general notation [68] deals with both equality and inequality constraints, depending on the lower and upper boundary values.

Once the aforementioned statements have been formulated, the *optimal control problem* aims to find the control profile $\mathbf{u}^*(t)$, in the space of all admissible controls U , which minimises the performance criterion J while respecting the differential model $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$ and the specified physical constraints. Briefly stated:

$$\begin{aligned} \min J &= \phi[t_f, \mathbf{x}(t_f)] + \int_{t_0}^{t_f} L[t, \mathbf{x}(t), \mathbf{u}(t)] dt, \quad u \in U \\ \text{subject to : } \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{g}_L^p &\leq \mathbf{g}^p[t(t), \mathbf{x}(t), \mathbf{u}(t)] \leq \mathbf{g}_U^p \\ \mathbf{g}_L^f &\leq \mathbf{g}^f[(\bar{t}_j), \mathbf{x}(\bar{t}_j), \mathbf{u}(\bar{t}_j)] \leq \mathbf{g}_U^f \end{aligned} \quad (7.6)$$

where the Bolza formulation is used to obtain the necessary conditions in the most general case.

7.2.4.1 Indirect Methods

Indirect methods are based on Pontryagin's maximum principle, adapting the sign convention for minimisation problem. This principle's derivation employs *calculus of variations* techniques, of which comprehensive references are [69] and [70]. The goal is to convert the *optimal control problem* as defined in the chapter's introduction into a *two-point boundary value problem* through the statement of the necessary conditions that a profile shall satisfy to be an optimal solution.

The process starts with the definition of an *augmented performance index* \bar{J} , in a fashion similar to equality-constrained static optimisation problems, where Lagrange's multipliers λ_j multiplying the dynamical constraints are summed to the objective function to form the *augmented performance index*:

$$\bar{J} = \Phi + \int_{t_0}^{t_f} \left[L[t, \mathbf{x}(t), \mathbf{u}(t)] + \boldsymbol{\lambda}^T(t) \{ \mathbf{f}[t, \mathbf{x}(t), \mathbf{u}(t)] - \dot{\mathbf{x}} \} \right] dt \quad (7.7)$$

According to the *calculus of variation*, the necessary conditions for a stationary extremum is that the first-order variation $\delta\bar{J}$ shall nullify at any instant of time for any constraint-allowed variation $\delta\mathbf{u}(t)$. The problem *Hamiltonian* is defined as:

$$H = L[t, \mathbf{x}(t), \mathbf{u}(t)] + \boldsymbol{\lambda}^T(t)\mathbf{f}[t, \mathbf{x}(t), \mathbf{u}(t)] \quad (7.8)$$

When path constraints are present, the Hamiltonian shall be augmented with the constraints' violation weighted by associated dual variables. After mathematical manipulation (see [67] for a detailed derivation), the necessary conditions for a control profile $\mathbf{u}^*(t)$ to be a stationary function of the performance index are represented by the following *Euler-Lagrange equations*:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \dot{\boldsymbol{\lambda}} &= -\left[\frac{\partial H}{\partial \mathbf{x}}\right]^T \\ \mathbf{0} &= \left[\frac{\partial H}{\partial \mathbf{u}}\right]^T \end{aligned} \quad (7.9)$$

where the relations in Eq. (7.9)-2 are labelled as *adjoint equations* and Eqs. (7.9)-3 as *control equations*. These differential equations, which a control profile has to necessarily satisfy to be a stationary solution, are coupled with a set of *transversality conditions*:

$$\begin{aligned} t_0 \text{ given} & \quad \vee \quad H(t_0) = 0 \\ t_f \text{ given} & \quad \vee \quad H(t_f) = -\frac{\partial \Phi}{\partial t} \Big|_{t_f} \\ \mathbf{x}(t_0) \text{ given} & \quad \vee \quad \boldsymbol{\lambda}(t_0) = 0 \\ \mathbf{x}(t_f) \text{ given} & \quad \vee \quad \boldsymbol{\lambda}(t_f) = \frac{\partial \Phi}{\partial \mathbf{x}} \Big|_{t_f} \end{aligned} \quad (7.10)$$

Hence, if any of the boundary conditions is a free parameter, either on time or state variables, the above conditions complete the minimum required number of known conditions at the initial or final time. Up to this point, the process defined the necessary conditions for a solution to be a stationary one. The *Legendre-Clebsch* condition about local convexity of the *Hamiltonian* shall be satisfied to ensure that the solution is an actual local minimum:

$$\frac{\partial^2 H}{\partial \mathbf{u}^2} \Big|_{\mathbf{u}^*} \geq 0 \quad (7.11)$$

The *TPBVP* defined by Eq. (7.9), coupled with the conditions (7.10) and (7.11), has no analytical closed-form solution for complex problems. Hence, numerical methods shall be employed. However, further information can be obtained by exploitation of the problem's first integrals. If the functions L and \mathbf{f} defined in the System (7.6) do not depend explicitly on the independent variable t , then the *Hamiltonian* is a first integral of the *TPBVP* along an optimal trajectory [71]. In general, if a first integral is found, the redundant information that it generates can be exploited to eliminate one *adjoint equation*, formally transforming the original *TPBVP* into another one of lower dimension, by following the procedure shown by Visser [67].

7.2.4.2 Direct Methods

A direct method does not require the derivation of the necessary conditions needed by indirect methods. On the contrary, it aims to find a sequence of profiles which progressively reduce the non-augmented performance index J and the constraint's violation. Direct methods require a parametrisation of the control functional form over trajectory's arcs. This is generally achieved by two conceptually different methods [71]:

- A grid at different times where the control parameters are to be found and the values within an interval are computed through interpolation.
- A set of orthogonal basis of mathematical functions dependent on time. Usually Fourier series, Legendre polynomials or the Chebyshev ones.

The goal is then to determine the values of the specified free parameters, either control values at fixed times in the grid form or the coefficients of the series in the second case, able to minimise the objective index and to respect the constraints. In this step, the number of free parameters is reduced from infinite degrees of freedom to a finite number of parameters, depending on the chosen parametrisation. This passage could seem a limitation of the direct methods when compared to the indirect ones. However, as already stated in the previous section, a numerical procedure is necessary also for indirect methods when dealing with complex cases such as low-thrust trajectory optimisation. These numerical methods require a so-called transcription to convert the infinite-dimension optimal problem into a solvable finite-dimension one. Hence, what seemed a limitation of the direct methods is a required passage of any technique nonetheless.

A direct method's solution is generally not an optimal solution itself, i.e. not a local minimum of the performance index, but just an approximation as a consequence of the discretisation or interpolation steps. Hence, the necessary conditions (7.9) and (7.11) can be used as an indicator of how close the found solution is to the real local optimum [72].

7.2.4.3 Comparison of Direct and Indirect Methods

Loosely comparing an optimal control problem to a static constrained optimisation, the direct method's goal is to pinpoint a local minimum of the performance function, while an indirect method aims to find a root of the necessary conditions. The latter shall be preferred when a closed-form solution is aimed for. Indeed, indirect methods allow to extract the control in an analytical way [73, 74]. However, this is possible only when several approximations are employed or simplified cases are considered. When a numerical approach is necessary, a direct method often results to be the simpler choice due to several considerations [68]:

- The quantities $\left[\frac{\partial H}{\partial \mathbf{x}}\right]^T$ and $\left[\frac{\partial H}{\partial \mathbf{u}}\right]^T$ needed by indirect methods must be analytically computed and changed when different models are employed. Furthermore, when a problem is divided into phases, these quantities change along the trajectory. This requires an extensive preliminary analytical stage for any different problem in the matter. On the contrary, a direct method is a flexible approach, more suitable for *black box* implementations, and able to handle a problem divided into different phases.
- Path inequalities, which are quite ordinary in low-thrust applications, represent a relevant issue for indirect methods. Indeed, a first guess of the *active-inactive* sequence is needed for practical methods as it changes the form of the Hamiltonian, by adding the *Lagrange multipliers*, the number of constrained arcs and the junction conditions. However, a priori knowledge of the right series is quite hard to achieve.
- Another issue with first guesses emerges from the initial estimate of the *adjoint variables* λ . As remarked by Bryson and Ho [75], the extremal solutions can be very sensitive to small changes in the unspecified boundary conditions. As usually the initial state variables are specified, the *transversality conditions* (7.10) show that the initial values of the adjoint variables for the optimal trajectory are not known. Further, these variables are not representing physical quantities. Hence, setting the right initial conditions, or even reasonable ones, is very complex, and a bad initialisation often results in numerically ill-conditioned solutions. On the contrary, direct methods disregard those variables and require only initial guesses on the physical state and control variables.

7.2.4.4 Practical Techniques for Optimal Control

As stated in numerous occasions, in general the continuous optimal control problem does not have a closed-form solution, and practical numerical optimisation methods come into play. Any numerical technique cannot handle an infinite-dimension problem, but it needs a discrete problem with a finite set of variables and constraints to work with. This transition can be performed with conceptually different methods which will be investigated in the present section. It is important to emphasise that

the following techniques are applicable to both indirect and direct approaches. A complete review of the common methods, with a focus on low-thrust trajectory optimisation, has been compiled by Betts [76], whereas in this section two major classes will be addressed.

Single Shooting

Typically, the single shooting method does not actually find application in the field of complex non-linear optimal control. However, it is useful to introduce the notation and several concepts shared by its extension, the *multiple shooting* method. The discretisation grid is composed by only two points, the initial and final times. Initially, the n free parameters in $\mathbf{y}^T = [\bar{x}_1, \dots, \bar{u}_{n_c}]$, composed by the initial conditions and the control parameters, are guessed. Hence, the trajectory is propagated forward (or equivalently backward) from the starting to the end time, leading to the final state:

$$\mathbf{x}_f^p = \mathbf{x}_0 + \int_{t_0}^{t_f} \mathbf{f}(t, \mathbf{x}, \mathbf{u}) dt$$

In general the propagated state \mathbf{x}_f^p will not coincide with the required final one \mathbf{x}_f . Hence, the difference between these two quantities becomes a constraint to nullify. In literature, this constraint is generally labelled as *defect*:

$$\mathbf{c}(\mathbf{y}) = \mathbf{x}_f^p - \mathbf{x}_f \quad (7.12)$$

The numerical values of the violation of the boundary conditions can be exploited to iteratively adjust the control parameters with NLP algorithms in order to finally solve the constrained minimisation.

The advantage of this basic method is that the NLP subproblem has only a small number of variables to optimise, i.e. the initial state guess and control parameters. However, for long time-scales and non-linear dynamics, even small changes in the parameters can result in very large defects change, leading to hypersensitivity with respect to the free parameters.

Multiple Shooting

In order to overcome the drawback of parameter sensitivity, it is possible to segment the overall time interval into a set of $m - 1$ smaller steps discretising the interval at m grid points $t_0 < t_1 < t_2 < \dots < t_f$. Then, each of the segments can be treated as an independent single shooting method, with continuity constraints added. Therefore, first guesses of the n_s state variables for each intermediate segment are now needed. The first guess trajectory is usually found by fast and low-fidelity methods. The

state variables at intermediate grid points are now control variables to be optimised. Hence, the number of control parameters in \mathbf{y} increases with respect to the single shooting method, precisely $n_y = (m - 1)(n_s + n_c \cdot n_p)$, where n_s is the number of state variables, n_c the control components and n_p the control parameters per each component. The *defect* equations can be expressed in the general form as:

$$\mathbf{c}(\mathbf{y}) = \begin{pmatrix} \mathbf{x}_2^p - \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_f^p - \mathbf{x}_f \end{pmatrix} \quad (7.13)$$

where again the goal is to nullify $\mathbf{c}(\mathbf{y})$. The dimension that the NLP subproblem shall solve, in order to link the different phases and minimise the objective function, dramatically increases with an increasing number of steps. However, the effects of changing a particular parameter are more intuitive for smaller steps, leading to an improvement of the convergence properties. In addition, the main drawback of the single shooting is solved, and, when the number of steps is high enough, the variables related to the first stages of the trajectory do not heavily influence the last phases. The segment decoupling mathematically translates into very sparse *Jacobian* and *Hessian* matrices, later involved by the NLP algorithm. For example, the Jacobian gets sparser and sparser as more phases are employed, because the percentage of non-zero elements is proportional to $1/(m - 1)$. This sparsity can be exploited to construct a computationally efficient non-linear programming subroutine, making the multiple shooting method both robust and competitive [77].

Collocation

The basic goal of *collocation* methods is to avoid repeated propagations over each segment. This is achieved by partitioning again the whole trajectory into $m - 1$ segments, leading to m grid points. Hence, the trajectory is only represented by the set of state variables $\mathbf{x}(t_k)$ and their derivatives $\mathbf{f}(t_k, \mathbf{x}(t_k), \mathbf{u}(t_k))$ at mesh points as well as the control profile nodes $\mathbf{u}(t_k)$. As these values are treated as NLP variables, gathered in the vector \mathbf{y} , the optimal control problem has been completely transcribed into a finite-dimensional NLP. For this reason, also collocation methods need a first guess solution, which can be sought with the aforementioned approaches. The state, state-derivative and control values within each interval are computed by interpolation through piecewise functions, usually Hermite (third order), Chebyshev or Lagrange polynomials (see [68] for detailed schemes) or Fourier series [78], whose coefficients depend on the adjacent grid points' state and derivatives. This a priori shape replaces the numerical integration process of shooting techniques with a much faster analytical propagation.

The differential equations $\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t))$ are substituted by a discretised form, which for a simple Euler scheme takes the following form:

$$\dot{\mathbf{x}} = \mathbf{f}(t_k, \mathbf{x}_k, \mathbf{u}_k) \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h} \quad (7.14)$$

where h is the interval size $t_{k+1} - t_k$. This Euler form is then transformed into a set of NLP constraints to be nullified:

$$c_k(\mathbf{y}) = \|\mathbf{x}_{k+1} - \mathbf{x}_k - h\mathbf{f}(t_k, \mathbf{y}_k, \mathbf{u}_k)\| \quad (7.15)$$

These constraints, which ensure the equation of motion to be approximately satisfied, are then coupled with the fixed-event ones and the path constraints, to construct a continuous trajectory and to respect the requested bounds at the grid points.

In collocation methods the choice of the interval size is vital because it influences the accuracy of the interpolated function in representing the true trajectory. An efficient procedure could be to compute initial estimates with a sparse grid and then refine it progressively. This implementation makes this technique very robust to imprecise initial guesses. Also in this method, the sparsity of the matrix shall be exploited as much as possible to make the algorithm efficient.

The greater drawback of collocation methods is that for problems dominated by highly non-linear dynamics, a very dense grid is needed to compute an accurate solution which, when integrated forward for validation, leads to small errors in the final state. This problem arises from the finite-difference approximation of the dynamics, as in Eq. (7.14) for an Euler scheme, and from the parametrisation of the shape. However, a dense grid translates into an expensive matrix inversion during the NLP subproblem, leading to the degradation of the computational performance.

Pseudospectral methods are a special class of direct collocation where the optimal control problem is transcribed by parameterising the state and control using global polynomials and collocating the differential-algebraic equations using the nodes obtained from a Gaussian quadrature [79, 80]. The terms pseudospectral and orthogonal collocation are used interchangeably in the literature.

7.3 Combinatorial and Network Optimisation

Until now, all optimisation problems and variables have been continuous, that is, each design vector has consisted of a set of variables composed of possible values within a specified range. In this section ‘combinatorial optimisation’ problems shall be discussed, where some or all of the design variables are restricted to a discrete set, most commonly binary integers, but also non-negative integers. This section will also discuss problems with a finite number of possibilities, namely, problems formulated to find a maximum or minimum of one or more functions, with many variables, which can be limited by a series of equality constraints, inequality constraints and bounds. All of these problems are ‘linear problems’ or can be reformulated as such with the inclusion of some constriction on one or more of the

design variables to ensure that only discrete values can be considered, described as ‘linear integer programs’. If the design vectors are pure (or all) integer, the problem is classed as a ‘pure integer program’, whereas if at least one (but not all) design vector integer, the problem is classed as a ‘mixed-integer program’, discussed in Sect. 7.3.2.1. Alternatively, these problems can be categorised into general non-negative integer problems or ‘binary’, where the discrete, integer design vectors hold a value of either 0 or 1. Most mixed-integer problems in practice are binary, and the use of integer variables, although beneficial in certain circumstances, is much less common.

Often in ‘combinatorial optimisation’, the phrases ‘combinatorial’, ‘discrete’ and ‘integer’ are used interchangeably with little explanation of the differences between them. Each of the three terms can be used to describe a problem or optimisation method formulated for use with integers, as opposed to continuous variables, as inputs and outputs of the problem or as components of the optimisation process. Often, the term **discrete**, when used to describe problems and processes, is used to simply describe the discrete nature of one or more aspects of said process, i.e. a ‘discrete problem’, as opposed to a continuous problem. It is not accurate to say that a discrete problem is always an integer problem, e.g. if a problem has discrete design variables $\mathbf{x} = 0, 0.3, 0.6, 0.9, 1.2$, the problem is considered ‘discrete’, but not integer. Similarly, the term **combinatorial** describes the problem formulation but can also be used to describe the origin or solution of a problem, and is categorised by the exponential explosion of variables or constraints, often modelled by ‘integer programming’. Finally, the phrase **integeri**, with respect to optimisation, is usually intended to describe the use of integer values in formulation or solution and thus also modelling. The similarities between these terms allows for a certain degree of interchangeability, though it is important to know where each term should and should not be used.

Integer programming was first recognised in the 1940s to 1950s, where the simplex algorithm (derived in 1948 and published in 1951) described a finite method suitable for application on any linear objective function subject to a finite set of linear constraints [81]. It was not until 1955 that Harold Kuhn derived a combinatorial algorithm for a single, specific integer problem using a dual-primal linear algorithm. Since then, a number of papers have expanded upon the available techniques and solving algorithms, introducing these tools to a wider and wider audience, such that many modern-day application rely on integer programming techniques.

Many real-world problems require the evaluation of integer problems; thus the interest in and knowledge applicable to optimisation of these problems are highly valued and ever-increasing. Many industries require the use of integer programming to solve practical problems. Communications, activity management, resource management, time scheduling and machine sequencing are vital to the cost minimising, resource management and time management of large commercial and industrial firms, whereas other problem applications are less grounded in real-life scenarios, such as high-energy physics and X-ray crystallography. These problems are generally more difficult to solve than problems that are linear and/or

continuous. Although the true advantage of combinatorial optimisation methods lies in the ability to process indivisible, discrete, real-life parameters, this optimisation category can also be utilised to convert continuous inputs to integer-only inputs with the intention of providing yes-no output values (which can be formulated as 0-1 integer problems), a particularly useful trait in machinery diagnosis.

Network optimisation is a special form of linear programming, where the structure of the program allows even faster solution approaches such as network simplex algorithm, and they are highly valued in their ability to optimise some of the most common, fundamental problems with minimal cost and a free flow of data to and from each network node. Analytically, network flow problems can solve some classes of combinatorial optimisation problems, such as shortest path, assignment and transportation. Network flow problems, although often complex, are utilised in the design and analysis of large connected problems, proving to be vital to the operation of many transportation, communication, manufacturing and social networks. Network optimisation methods make use of powerful techniques such as data caching, streamlining of data protocols and even data elimination. These techniques, when correctly applied, can assist in developing faster data transfers, accurate transport solutions and improved response times for software applications.

7.3.1 *Pure Integer Optimisation*

As introduced earlier, integer-only programming is a form of combinatorial optimisation developed for cases where all design vectors are integer, i.e. $\mathbf{x} \in \{0, 1, 2 \dots\}$. Mathematically speaking, the original problem formulation, shown in the Introduction section of this chapter, can be altered to describe the case of an integer-only problem. Integer-only or ‘pure integer’ optimisation can make use of combinatorial optimisation algorithms since the search space, and hence the number of potential solutions is finite. Moreover, in a constrained integer-only problem, the number of potential solutions is limited by the number of possible combinations of every integer. For smaller problems, an exhaustive search may be used to evaluate each point in the design space, but this cannot be extended to larger problems due to the curse of dimensionality. This can be seen visually by Fig. 7.1, a simple integer-only problem with two integer inputs and three linear inequality constraints, as formulated in Eqs. (7.16) to (7.19).

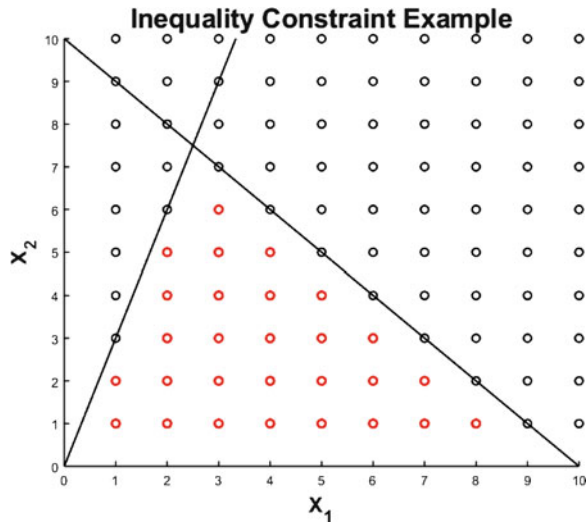
$$\mathbf{x} = [x_1 \ x_2] \in \{0, 1, 2 \dots\} \quad (7.16)$$

$$A = [3 \ -1] \quad (7.17)$$

$$b = [0 \ -10] \quad (7.18)$$

$$lb = [0 \ 0]; \quad ub = [10 \ 10]; \quad (7.19)$$

Fig. 7.1 Integer problem with linear constraints



This can be seen graphically in Fig. 7.1. We note that the number of points available for evaluation by the function are extremely limited and clearly seen as finite and by extension, the number of different values provided by the function is finite. This is due to the small number of dimensions. As the number of dimensions increases, the number of possible points increases exponentially. For simple problems such as this, an exhaustive search can be used. Complications arise upon the introduction of additional dimensions of the design vector where the size of the search space increases drastically depending upon the bounds, as described by the ‘curse of dimensionality’.

7.3.1.1 Special Case: 0-1 Integer Programming

As introduced earlier, 0-1 integer programming or ‘binary programming’, is a special category of integer-only problem where the design variables are either one or zero (binary). This problem formulation is most commonly used for decision-making, when the inputs to a function is one of only two possible values: yes/no, open/closed, true/false, etc. [82].

Its mathematical formulation is

$$\sum_{j=1}^n c_j^T \mathbf{x}_j \tag{7.20}$$

subject to: $A_{i,j} \mathbf{x}_j \leq b_i$ (7.21)

$\mathbf{x}_i = \{0, 1\}$ (7.22)

Many real-life decision-making problems involve significant yes/no decisions, most often at strategic and tactical levels. The knapsack problem is a classic example of this type of problem. Moreover, other non-binary problems can be converted into this form if beneficial, where values can be split into ones and zeros to represent high/low temperatures, fast/slow speeds and other extremes. This can be added in practice by introducing equality constraints such that $0 = \mathbf{x}$ and $1 = \mathbf{x}$. Take a condition monitoring system which uses a temperature input to determine if a particular part has overheated. Binary programming may be applied where temperatures above a certain value are considered a failure (1) and temperatures below this are considered acceptable (0).

7.3.2 *Mixed-Integer Programming*

Mixed-integer programming, although commonly utilised in many modern-day problems, was developed following the formulation of the simplex method, developed by Dantzig in 1951. This was followed by the work from Ford and Fulkerson, whose earliest contribution to network flow began with ‘maximal flow through a network’, which is often credited as the original algorithm designed to solve maximum flow problems, and thus is considered one of the most influential papers in the development of further algorithms used for solving and analysing network flow models [83]. The simplex method also gave way to the first pure integer optimisation algorithm developed by Gomory in 1958. The increase in complexity resulted in an increase in computational cost, best modelled by a polynomial-time algorithm. These models allowed for the categorisation of problems into categories depending on hardness, where integer programming was considered NP-hard in general.

More complex problems may require the use of mixed-integer variables. That is to say that one or more variables of function $f(\mathbf{x}_{m+n})$ are a set of continuous variable(s), $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$, and other variable(s) that are discrete values $\mathbf{x}_{m+1}, \mathbf{x}_{m+2}, \dots, \mathbf{x}_{n+m}$.

Mixed-integer problems cannot be solved by a continuous variable-based solver, as the step size, Δx , may be unsuitable for discrete variables. Take the steepest descent algorithm as an example: if the step size, λ , is not compatible with the design variables (in this case, not an integer), this will result in the failure of the algorithm and thus, no solution. Conversely, discrete solvers may disregard step sizes for continuous variables that are smaller than one, thus increasing the inaccuracy in the solver. This acts as a form of proof of the NFL theorem but also highlights the need for the development and correct application of more integer programming methods and categories, such as linear and non-linear.

7.3.2.1 MIP vs MINLP

Mixed-integer programming can be further categorised into mixed-integer programming (MIP) and mixed-integer non-linear programming (MINLP). Problems can be categorised by the nature of the objectives and constraints with respect to the design vector. Mixed-integer programming follows the standard linear programming formulation, where the objectives and constraints are linear with respect to the design variables which, in this case, consist of at least one design parameter composed of integers ($0, 1, 2 \dots n$) and at least one design parameter composed of continuous values, described in Eq. (7.23).

$$\text{Minimise } c^T(x) \quad (7.23)$$

where

$$Ax \geq b$$

$$x \geq 0$$

$$x_j \in Z \quad \forall j \in I$$

Commercial solvers such as IBM Ilog Cplex, FICO Xpress and Gurobi as well as open-source solvers such as COIN-OR all employ the branch-and-bound algorithm at their core, where the solution is iteratively parted into smaller subproblems, most commonly referred to as the left and right child problems (with respect to the original problem), discussed further in Sect. 7.3.2.2. Mixed-integer problems can also be solved by iteratively solving the so-called separation problem, where the feasible region of the problem is cut off by adding valid ‘cuts’ (i.e. additional constraints) and hence by elimination of sections of the design space. This is commonly known as the ‘cutting plane algorithm’. Where the branch-and-bound algorithm employs LP relaxation to simplify the subproblems, the cutting plane algorithm tightens the LP relaxation to find a better approximation of the convex hull. All MIP solvers employ the so-called branch-and-cut method, which combines these two major solution methods for a more effective solution process.

A mixed-integer non-linear programming problem consists of at least one design parameter composed of discrete integers ($0, 1, 2 \dots n$) and at least one design parameter composed of continuous values, similar to MIP. However, in this case, either the objective or at least one of the constraints is non-linear with respect to design vectors. These problems follow the form:

$$\text{Minimise } f(\mathbf{x}) \quad (7.24)$$

where

$$A_i(\mathbf{x}) = 0 \quad \forall i \in E \quad (7.25)$$

$$b_i(\mathbf{x}) \leq 0 \quad \forall i \in I \quad (7.26)$$

$$\mathbf{x} \in \{x_1, x_2 \dots x_{n+m}\} \quad (7.27)$$

where $x_1, x_2 \dots x_n$ are integer variables and $x_{n+1}, x_{n+2} \dots x_{n+m}$ are continuous variables. These problems are particularly complex, combining the combinatorial difficulty of integer optimisation with non-linear functions, and so few algorithms have been designed specifically for use with these problems. Most solution algorithms for MINLPs fall into one of the two categories: single-tree and multi-tree methods [84]. MINLPs can be solved using a generalised Benders' decomposition (multi-tree), where a general MIP problem is employed with non-linear programming subproblems, and a modified branch-and-bound method (multi-tree), where modifications are made to improve the performance of the algorithm. Complications arise during the optimisation of a non-convex MINLP since even after relaxation of the integer design variables to continuous design variables, the function can remain non-convex, resulting in many local minima.

Many MINLP methods break the problem into their MIP and NLP components and solve the overall problem iteratively.

7.3.2.2 Methods

The algorithm applied to solve a combinatorial problem is dependent largely on the possible formulation of the problem and the requirements of the user. As per the 'no free lunch' theorem, a single algorithm cannot find the best possible solution to all possible problems. To find the most suitable solution method, the basic problem formulation must be considered: linear/non-linear, small/large, single-objective/multi-objective and binary/integer/mixed-integer.

Similarly to integer-only and continuous-only programming, combinatorial problems can be categorised as linear and non-linear. Also similarly to integer-only and continuous-only problems, linear problems are typically less complex than non-linear problems. As a result, fairly large, moderately complex problems can be solved using 'exact' methods, where every part of the problem and its subproblems are solved either explicitly or implicitly. For larger, more complex problems, 'heuristic' methods may be used. Heuristics use intuitive techniques to find a 'rough' solution to any given problem to a certain degree of accuracy.

Exact Methods

With relatively simple problems with low computational costs, exact methods can be used to solve combinatorial problems. These methods, unlike heuristic methods described in Sect. 7.3.2.2, guarantee an optimal solution and thus are the ideal choice. These solution methods can be placed into one of the two categories: implicit enumeration or explicit enumeration. Explicit solution methods are often

the simplest way of solving a problem using all possible solutions (an extensive search), but due to the ‘curse of dimensionality’, the complexity of the problem rises exponentially with the increasing number of dimensions within the design vector. As such, explicit enumeration is a valuable tool for small integer problems where dimensionality is limited. For larger problems, implicit enumeration is used.

Within explicit enumeration, the optimisation algorithm builds all the possible solutions to find the optimal solution. This is typically more costly than implicit enumeration, where all possible solutions are considered in some manner without explicit evaluation. Implicit enumeration methods consist of a wide variety of possible optimisation algorithms, where the most common are ‘divide-and-conquer’ methods, where a problem is divided into sets of m groups of problems iteratively until a subproblem is simple enough to be solved, and the branch-and-bound method, as discussed below.

Branch and Bound

In 1960, Alison Doig and Ailsa Land published a paper entitled ‘An Automatic Method for Solving Discrete Programming Problems’, introducing the concept of branch-and-bound algorithms. Although first intended to solve combinatorial optimisation problems, many improvements have been made to generalise the algorithm to solve continuous problems and improve the efficiency.

When solving MIP problems, the branch-and-bound method does not consider integer design variables as discrete values but rather converts these discrete values to continuous values by relaxation of the integer restrictions. This simplifies manipulation of the problem and thus, decreases the difficulty to solve.

The ‘branch-and-bound’ method consists generally of three main techniques: branching, bounding and searching.

- Branching
 - This step splits the continuous search space into several smaller subspaces, eliminating infeasible parts of the continuous space through application of necessary conditions for integer solutions.
- Bounding
 - The method of bounding depends on whether the objective function is to be maximised or minimised. If this function is to be maximised, an upper bounding strategy is used, and if minimised, a lower bounding strategy is applied.
- Searching
 - The process of searching each subspace for an optimal solution, preferably the most promising region first.

To begin the branching process, the search space S is split into a number of smaller and mutually disjoint subsets S_1, S_2, \dots, S_r . Following this partition, each subspace is analysed to find a local, feasible minimum, where each subset is also a set of feasible solutions of a ‘candidate problem’, which is found by imposing

additional constraints on the original function. The search space thought to contain the ‘best solution’ is then analysed. If the optimal solution is found, the subspace, and thus also the candidate problem, is fathomed. If not, the problem only contains a lower bound for the minimum objective value in it, and this subspace is divided into yet smaller subspaces (or candidate problems), and the process is repeated. This process can be adapted for specific problems. Consider problems with 0-1 variables. To branch these problems, extra constraints can be added to constrict $x_1 = 0$ or $x_1 = 1$, creating two candidate subproblems. In this case, x_1 is known as the branching variable

The ‘bound’ step of the branch-and-bound algorithm is dependent on the objective of the objective function. Assuming that the objective function z is to be minimised, lower bounding strategies are required. Any lower bounding strategy should be simple, efficient and run with a low computational cost. In any case, a lower bounding method should bound closest to the minimum value of z , which can be handled using one of the many strategies [85].

- **Relaxation of constraints:** All difficult or computationally costly constraints are relaxed, and z is minimised for only the remaining constraints. Using this method, the minimum value of z is equal to the lower bound for z_{min} in the original problem.
- **Modification of the objective function:** In this case, the modified objective function is created such that $f \leq z$ for all feasible solutions. Furthermore, f should be easy to minimise subject to the original constraints. Subject to these properties, f is a lower bound to z_{min} of the original problem.
- **Lagrangian Relaxation:** A Lagrangian multiplier is created where u in $L(u, x)$ is associated with the relaxed constraints. In this case, the optimum z is a lower bound of z_{min} of the original problem.
- **Branch-and-cut:** Otherwise known as ‘cutting planes’, this iterative method solves the LP relaxation at each solution, and depending on if the solution is optimal or not, it is either accepted (if optimal) or a linear constraint is found that excludes the LP solution and no others. This constraint is referred to as a ‘cut’.

The branch-and-bound algorithm is searched using a ‘search tree method’. In this case, the original solution is analysed and branched, splitting the problem into two candidate problems. These two problems are bound, analysed and branched. Candidate problems which do not contain an optimal solution are not branched and become terminal nodes. Candidate problems which contain an optimal solution are further branched, and the process repeats. Terminal nodes may be required in further iterations of the algorithm. This search method continues to branch until an optimal solution is found.

Since its introduction in 1960, the branch-and-bound method has been slightly altered for improved results on specific problems. One such example of this is the ‘Beale and Small’ method [86]. This method uses a different bounding strategy, includes the termination of particularly non-optimal subspaces and includes a heuristic ‘worst alternative’ branching method.

For any strategy, the lower bound must be fairly close to the minimum objective value and generate candidate problems where the lower bounds are as high as possible. The computational time of the strategy, including calculation of the lower bounds for every candidate problem, must be low enough that the algorithm can be iterated many times.

Heuristic Methods

Methods to solve combinatorial optimisation problems discussed so far have been exact, i.e. finding the global solution is guaranteed (if there is a feasible solution). When heuristics are involved, this is not the case. Heuristics provide alternative methods for finding solutions to challenging problems (in particular in real-world settings) that do not guarantee an optimal solution (some of them only in statistical way). We note that this is different than approximation algorithms, which provide a performance guarantee such as maximum deviation from the optimal solution. Conversely to deterministic sampling, heuristic sampling requires a distribution of sample points over the search space with a higher density of points in areas of particular interest. Common heuristics include (a) relaxation-based heuristics and (b) rounding-based heuristics.

These methods are valid for only convex or small-scale non-convex MINLPs. There is no method yet that can reliably solve large-scale MINLPs, and, when compared, the algorithms that exist to solve convex MINLPs do not show a clear ‘best algorithm’, as can be expected. Where MINLP algorithms lack in computational speed and other desirable characteristics, a mixed-integer problem (MIP) is often used as a replacement for large-scale, real-world problems. Even without non-linear constraints, these problems can still be extremely hard, actual NP-hard [87].

The nearest neighbour heuristic and the Christofides algorithm [88] are well-known start heuristics for the TSP. The k-OPT-algorithm is an improvement heuristic which was originally designed for the TSP, but variants of this are used for several other combinatorial optimisation problems. It also formed the basis for the Lin-Kernighan heuristic [89] which is one of the most common algorithms used to find good solutions for TSPs. Balas and Martin [90] presented the pivot-and-complement that was developed for binary programs (BPs) and is based on the observation that, in the nomenclature of the simplex algorithm, an LP-feasible solution of which all basic variables are slack variables is also integer feasible. It performs pivot operations which drive the integer variables out of the basis and the slacks into the basis. The same authors [91] developed another method called pivot-and-shift that can be applied to general MIPs. The method was further improved with more pivot types and new rules for selecting them, as well as an extension of the shifting procedure, and a neighbourhood search related to local branching [92].

Another method is the so-called heuristic ceiling point algorithm, which was restricted to integer problems (IPs) without equality constraints. Scatter search with star paths is a diversification heuristic [93] that creates a couple of points which are

then linked by paths along which feasible solutions are searched. The main goal of Scatter search is to diversify the set of solutions and not improving the incumbent.

The Octahedral Neighbourhood Enumeration (OCTANE) search is a heuristic for BPs based on a ray shooting algorithm starting at the LP-optimum and hitting the facets of the octahedron dual to the unit hypercube [94].

In more recent years, some large neighbourhood search heuristics have been presented, such as the local branching [92] and the relaxation Induced Neighborhood Search (Rins) [95].

7.3.3 *Network Optimisation*

Network optimisation is a special type of linear programming, where variables are represented as flows in a network. Many real practical problems can be formulated as a ‘network optimisation’ problem, most commonly very large problems including the study of traffic, train and population flow, distribution analysis and communication problems. Consequently, many optimisation non-specialists understand the importance of these optimisation algorithms, which led to the widespread use of network optimisation in the testing and devising of new theories. This problem-solving method can be used to solve a series of combinatorial problems, for example [96, 97]:

- Space-time networks [98]
 - Traffic flow simulating, airline scheduling [85]
- Physical networks
 - Designing of streets and pipelines [99] to best manage flow
- Route networks
 - Vehicle route flows, map route optimisation (e.g. bus routes) [100]
- Constructing matches
 - Bipartite matching, survey design

Please note that problems such as TSP, VRP and scheduling may be represented using a network, but that does not mean they are network optimisation problems. Network optimisation is still LP and does not contain any integer variables; then the problems can be solved very effectively.

Standard Network Flow Formulation and Notation

A typical ‘network’ is a series of nodes (or vertices) connected by arcs (or edges), where each node is associated with a new design value and each arc is associated

with some category of moving value. To minimise unnecessary problem evaluations, it is assumed that no points are part of any ‘self-loop’, and so an arc from one point cannot lead back to the same point. A problem with many arcs and/or edges can be categorised into one of three forms: ‘directed network’, where only arcs are present; ‘undirected network’, where only edges are present; or ‘mixed network’ if there is a combination of arcs and edges [85]. In literature, arcs and edges are often considered as the same entity, and, in the following, they will be generally referred to as lines.

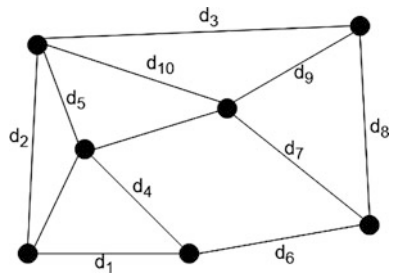
This problem formulation can be optimised for different specific problem requirements. This option to include specification of algorithm type allows for maximisation of accuracy and efficiency, as per the NFL theorem. Problems best suited for network optimisation include:

- Shortest path problem
 - Shortest path problems are some of the most commonly encountered network optimisation problems both in transportation and in communication.
- Maximum flow problem (as discussed)
 - Find a feasible flow path from a single source to a single sink, such that the flow is maximised.
- Minimum weight spanning tree
 - This problem requires each node to connect to every other node. If the links between nodes are expensive, it may be desirable to have each node connect to only two other nodes.

This can be formulated mathematically by considering a graph or directed network $G = (N, A)$, where N is a series of nodes (otherwise known as ‘points’ or ‘vertices’) such that $N = \{1, 2, 3 \dots m\}$ and A is a series of lines $A = \{a_1, a_2, a_3 \dots a_n\}$ [96, 101], with a cost $c_{i,j}$ and a capacity associated with every line or arc $(i, j) \in A$. These problems can be shown pictorially by placing all nodes and connecting lines on a plane, as can be seen in Fig. 7.2.

This problem can also be described mathematically using a graph. Unless otherwise specified, it can be assumed that the edges are distinct such that if $a = (i, j)$ then $i \neq j$, this would generate a ‘simple’ graph. Many network problems can be formulated in this way, where N could be a set of locations, A a set of

Fig. 7.2 Visual representation of a network



potential routes between these cities and c a set of distances. Typically, network problems are bound by both flow constraints and flow bounds. The upper flow bound or ‘capacity’ of an arc is denoted commonly by k_{ij} , describing the maximum possible quantity of material that can be moved across each node.

$$l_{ij} \leq f_{ij} \leq k_{ij} \quad (7.28)$$

For any given problem, there is a ‘balance’ constraint for each node, where basically the net flow from this node (i.e. outflow-inflow) will be equal to the ‘supply’ of this node. The supply b_i of node i is either positive (e.g. if this node is a location providing entries into the network), negative (e.g. if this node is a client with a demand) or zero (if the node plays a location for transshipment). Then, the balance constraint will be in the following form [102]:

$$\sum_{j \in N} f_{ij} - \sum_{j \in N} f_{ji} = b_i \quad \text{Flow Balance Equations} \quad (7.29)$$

Provided the network follows these bounds and constraints, and the supplies of the nodes are balanced, the network will be valid.

7.4 Summary

This chapter gives a brief introduction to optimisation problem formulations and solution methods. After a general overview of different problems, the chapter is mainly divided in two main sections: continuous problems and methods and discrete problems and methods.

The first section on continuous problems is further divided into four parts, related to local methods, optimal control, global methods and multi-objective optimisation. On the other hand, the section on discrete problems is composed by three parts on pure integer optimisation, mixed-integer optimisation and network optimisation.

This chapter is meant to give an accessible introduction to formulation and solving methods. The reader is kindly invited to use the list of references and read the following chapters, to know more about the methods and to see what are the most recent advances in the field.

References

1. D.H. Wolpert, W.G. Macready, *No Free Lunch Theorems for Optimization* (IEEE, Piscataway, 1997)
2. J. Nocedal, S.J. Wright, *Numerical Optimisation* (Springer, Berlin, 1999)

3. J. Knowles, D. Corne, *Memetic Algorithms for Multiobjective Optimization: Issues, Methods and Prospects* (IEEE Press, Piscataway, 2000), pp. 325–332
4. J.D. Pinter, *Global Optimization in Action* (Springer, Berlin, 1996)
5. R. Horst, P.M. Pardalos, H.E. Romeijn, *Handbook on Global Optimization: Nonconvex Optimization and Its Applications* (Springer, Berlin, 1995)
6. S. Rajasekaran, On simulated annealing and nested annealing. *J. Glob. Optim.* **16**, 4356 (2000)
7. M. Locatelli, Simulated annealing algorithms for continuous global optimization. *J. Optim. Theory Appl.* **104**, 121–133 (2000)
8. T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms* (Oxford University Press, Oxford, 1996)
9. C.A. Floudas, *Deterministic Global Optimization* (Springer, Berlin, 2000)
10. P.M. Pardalos, Enumerative techniques for solving some nonconvex global optimization problems. *OR Spektr.* **10**, 29–35 (1988)
11. W. Forster, *Homotopy Methods*. Handbook of Global Optimization: Nonconvex Optimization and Its Applications (Kluwer, Dordrecht, 1995), pp. 669–750
12. I. Diener, *Trajectory Methods in Global Optimization*. Handbook of Global Optimization: Nonconvex Optimization and Its Applications (Kluwer, Dordrecht, 1995), pp. 649–668
13. R. Horst, H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd edn. (Springer, Berlin, 1996)
14. T.W. Simpson, J. Peplinski, P.N. Koch, J.K. Allen, Metamodels for computer-based engineering design: survey and recommendations. *Eng. Comput.* **17**(2), 129–150 (1995)
15. E.R. Hansen, G.W. Walster, *Global Optimization Using Interval Analysis* (CRC Press, Boca Raton, 2003)
16. A.V. Levy, S. Gomez, The tunneling method applied to global optimization, in *Numerical Optimization* (SIAM, Philadelphia, 1985), pp. 213–244
17. F.J. Solis, R.J.-B. Wets, Minimization by random search techniques. *Math. Oper. Res.* **6**, 19–30 (1981)
18. H.J. Kushner, A versatile stochastic model of a function of unknown and time varying form. *J. Math. Anal. Appl.* **9**, 379–388 (1962)
19. J. Mockus, On bayesian methods of optimization, in *Toward Global Optimization*, ed. by L.C.W. Dixon, G.P. Szegö (North Holland, Amsterdam, 1975)
20. G. Rudolph, Convergence of evolutionary algorithms in general search spaces, in *IEEE International Conference on Evolutionary Computation* (1996), pp. 50–54
21. G. Rudolph, Evolutionary search under partially ordered fitness sets, in *International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems (ISI 2001)* (2001), pp. 818–822
22. A.H.G. Rinnooy Kan, G.T. Timmer, Stochastic methods for global optimization. *Am. J. Math. Manag. Sci.* **4**, 7–40 (1984)
23. A.H.G. Rinnooy Kan, G.T. Timmer, Stochastic global optimization methods, part I: clustering methods. *Math. Program.* **39**, 27–56 (1987)
24. A.H.G. Rinnooy Kan, G.T. Timmer, Stochastic global optimization methods, part II: multi level methods. *Math. Program.* **39**, 57–78 (1987)
25. J.M. Laarhoven Peter, H.L. Aarts Emile, *Simulated Annealing* (Springer, Berlin, 1987)
26. M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, 1998)
27. M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, in *5th International Conference on Parallel Problem Solving from Nature (PPSN V)*. Lecture Notes in Computer Science, vol. 1498 (Springer, Berlin, 1998), pp. 418–427
28. P. Larrañaga, R. Etxeberria, J. Lozano, J. Peña, Optimization in continuous domains by learning and simulation of Gaussian networks, in *Conference on Genetic and Evolutionary Computation (GECCO00) Workshop Program*, pp. 201–204. (Morgan Kaufmann, San Mateo, 2000)

29. M. Costa, E. Minisci, MOPED: a multi-objective Parzen-based estimation of distribution algorithm for continuous problems, in *Evolutionary MultiCriterion Optimisation 2003*. Lecture Notes in Computer Science, vol. 2632 (Springer, Berlin, 2003), p. 71
30. P. Larrañaga, H. Karshenas, C. Bielza, R. Santana, A review on probabilistic graphical models in evolutionary computation. *J. Heuristics* **18**(5), 795–819 (2012)
31. K.V. Price, R.M. Storn, J.A. Lampinen, Differential evolution, in *A Practical Approach to Global Optimization, Natural Computing Series* (Springer, Berlin, 2005)
32. M. Vasile, E. Minisci, M. Locatelli, An inflationary differential evolution algorithm for space trajectory optimization. *IEEE Trans. Evol. Comput.* **15**(2), 267–281 (2011)
33. D.J. Wales, J.P.K. Doye, Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J. Phys. Chem. A* **101**, 5111–5116 (1997)
34. B. Addis, M. Locatelli, F. Schoen, Local optima smoothing for global optimization. *Optim. Methods Softw.* **20**, 417–437 (2005)
35. S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **15**(1), 4–31 (2011)
36. J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm. *Soft Comput. A Fusion Found. Method. Appl.* **9**(6), 448–462 (2005)
37. A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **13**(2), 398–417 (2009)
38. M.M. Ali, A. Trn, Population set based global optimization algorithms: some modifications and numerical studies. *Comput. Oper. Res.* **31**(10), 1703–1725 (2004)
39. J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**(6), 646–657 (2006)
40. E. Minisci, M. Vasile, Adaptive inflationary differential evolution, in *Congress on Evolutionary Computation (CEC2014)*, July 6–11, Beijing (2014)
41. M. Di Carlo, M. Vasile, E. Minisci, Multi-population adaptive inflationary differential evolution algorithm with adaptive local restart, in *Congress on Evolutionary Computation (CEC2015)* (2015)
42. M. Clerc, *Particle Swarm Optimization* (ISTE, London/Newport Beach, 2006)
43. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4 (1995), pp. 1942–1948
44. K. Miettinen, *Nonlinear Multiobjective Optimization* (Springer, Berlin, 1999)
45. C.A. Coello Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowl. Inf. Syst.* **1**, 269–308 (1998)
46. C.M. Fonseca, P.J. Fleming, An overview of evolutionary algorithms in multiobjective optimization. *Evol. Comput.* **3**(1), 1–16 (2007)
47. J. Brian, J. Ritzel, E. Wayland, S. Ranjithan, Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resour. Res.* **30**(5), 1589–1603 (1994)
48. Y. Ijiri, *Management Goals and Accounting for Controls* (North-Holland Publishing Company, Amsterdam, 1965)
49. Y. L. Chen, C.C. Liu, Multiobjective VAR planning using the goal attainment method. *IEE Proc. Gener. Transm. Distrib.* **141**(3), 227–232 (1994)
50. L.A. Ricciardi, C.A. Maddock, M. Vasile, Direct solution of multi-objective optimal control problems applied to spaceplane mission design. *J. Guid. Control. Dyn.* **42**(1), 30–46 (2019)
51. M. Vasile, Multi-objective optimal control: a direct approach, in *Satellite Dynamics and Space Missions*, ed. by G. Baú, A. Celletti, C. Gales, G. Federico Gronchi (Springer, Berlin, 2019)
52. D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Boston, 1989)


53. C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, in *Genetic Algorithms Proceedings of the Fifth International Conference*, San Mateo (1993), pp. 416–423
54. N. Srinivas, K. Deb, Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **2**(3), 221–248 (1994)
55. J. Horn, N. Nafpliotis, Multiobjective optimization using the Niche Pareto Genetic algorithm, IlliGAL Report n.93005, University of Illinois (1993)
56. E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)
57. E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, TIK-Report 103 (2001)
58. K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist Multi-Objective Genetic Algorithm: NSGA-II, KanGAL Report No. 200001 (2000)
59. J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto Archived Evolution Strategy. *Evol. Comput.* **8**(2), 149–172 (2000)
60. J.D. Knowles, D.W. Corne, M.J. Oates, *The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization*. Lecture Notes in Computer Science (2000), pp. 839–848
61. D.W. Corne, N.R. Jerram, J.D. Knowles, M.J. Oates, PESA-II: region based selection in evolutionary multiobjective optimization, in *Proceedings of the Genetic and Evolutionary Computation Conference* (2001)
62. C.A. Coello Coello, G. Toscano Pulido, *A Micro-Genetic Algorithm for Multiobjective Optimization*. Lecture Notes in Computer Science (2001), pp. 126–140
63. C.A. Coello Coello, G. Toscano Pulido, *The Micro Genetic Algorithm 2: Towards On-Line. Adaptation in Evolutionary Multiobjective Optimization*. Lecture Notes in Computer Science (2003), pp. 75
64. P. Czyzak, Pareto Simulated Annealing, a meta-heuristic technique for multiple objective combinatorial problems. *J. Multi-Criteria Decis. Anal.* **7**(1), 34–47 (1998)
65. C.A. Coello Coello, G. Toscano Pulido, M.S. Lechuga, Handling multiple objectives with Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **8**, 256–279 (2004)
66. K. Socha, M. Dorigo, Ant colony optimization for continuous domains. *Eur. J. Oper. Res.* **185**(3), 1155–1173 (2008)
67. H.G. Visser, Aircraft Performance Optimization. Delft University of Technology (2014)
68. J. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, 1st edn. (Society for Industrial & Applied Mathematics, Philadelphia, 2001)
69. H. Goldstein, *Classical Mechanics*, 3rd edn. (Pearson, London, 2001)
70. G. Bliss, *Lectures on the Calculus of Variations*, 1st edn. (University of Chicago Press, Chicago, 1946)
71. S. Kemple, *Interplanetary Mission Analysis and Design*, 1st edn. (Springer, Berlin, 2006)
72. B.A. Conway, *Spacecraft Trajectory Optimization* (Cambridge University Press, New York, 2010)
73. R.H. Bishop, D.M. Azimov, Analytical space trajectories for extremal motion with low-thrust exhaust-modulated propulsion. *J. Spacecr. Rocket.* **38**(6) (2001)
74. J.A. Kechichian, Optimal low-thrust transfer using variable bounded thrust. *Acta Astronaut.* **36**(7) (1995)
75. A. Bryson, Y. Ho, *Applied Optimal Control* (John Wiley & Sons, Hoboken, 1975)
76. J. Betts, Survey of numerical methods for trajectory optimization. *J. Guid. Control. Dyn.* **21**(2), 193–207 (1998)
77. C. Greco, Variational multiple shooting: theory and applications. Delft University of Technology (2017)
78. F. Zuiani, M. Vasile, Direct transcription of Low-Thrust trajectories with finite trajectory elements, in *61st International Astronautical Congress*, Prague (2010)
79. C. Canuto, M.Y. Hussaini, A.M. Quarteroni, T.A. Zang, *Spectral Methods in Fluid Dynamics*. Springer Series in Computational Physics (Springer, Berlin, 1988)

80. B. Fornberg, *A Practical Guide to Pseudospectral Methods* (Cambridge University Press, Cambridge, 1998)
81. M. Jnger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, L.A. Wolsey, *50 Years of Integer Programming 1958–2008: From the Early Years to the State-of-the-Art* (Springer, Berlin, 2009)
82. A. Kaufmann, A. Henry-Labordère, *Integer and Mixed Programming: Theory and Applications*, vol. 137 (Elsevier, Amsterdam, 1977)
83. M. Josefsson, M. Mützell, Max Flow Algorithms Ford-Fulkerson, Edmond-Karp, Goldberg-Tarjan Comparison in regards to practical running time on different types of randomized flow networks. KTH Computer Science and Communication, Stockholm (2015)
84. P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, Mixed-integer nonlinear optimization. *Acta Numer.* **22**, 1–131 (2013)
85. K. Murty, *Linear and Combinatorial Programming* (John Wiley & Sons, Inc., New York, 1976)
86. J. Abadie, *Integer and Nonlinear Programming* (North-Holland Pub. Co., Amsterdam, 1970)
87. A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency* (Springer, Berlin, 2003)
88. N. Christofides, Worst-case analysis of a New Heuristic for the travelling salesman problem. GSIA report 388, Carnegie-Mellon University (1976)
89. S. Lin, B.W. Kernighan, An effective heuristic algorithm for the travelling-salesman problem. *Oper. Res.* **21**, 498–516 (1973)
90. E. Balas, C.H. Martin, Pivot-and-complement: a heuristic for 0-1 programming. *Manag. Sci.* **26**(1), 86–96 (1980)
91. E. Balas, C.H. Martin, *Pivot-and-Shift: A Heuristic for Mixed Integer Programming* (GSIA, Carnegie Mellon University, Pittsburgh, 1986)
92. M. Fischetti, A. Lodi, Local branching. *Math. Program.* **98**(1), 23–47 (2003)
93. F. Glover, A. Lkktangen, D.L. Woodruff, Scatter search to generate diverse MIP solutions, in *OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research* (2000)
94. E. Balas, S. Ceria, M. Dawande, F. Margot, G. Pataki, Octane: a New Heuristic for pure 0-1 programs. *Oper. Res.* **49**(2), 207–225 (2001)
95. E. Danna, E. Rothberg, C.L. Pape, Exploring relaxation induced neighborhoods to improve MIP solutions. *Math. Program.* **102**, 71–90 (2005)
96. G.L. Nemhauser, A.G.H. Rinnooy Kan, M.J. Todd, *Optimization*, vol. 1 (North Holland, Amsterdam, 1989)
97. D. Bertsekas, *Network Optimization: Continuous and Discrete Models* (Athena Scientific, Belmont, 1998)
98. N. Shah, S. Kumar, F. Bastani, I.L. Yen, A space-time network optimization model for traffic coordination and its evaluation, in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing* (SUTC 2008), Taichung (2008), pp. 177–184
99. South Staffs Water, *Network Optimisation and Energy Management Business Strategy* (2013)
100. C. Sun, L. Cheng, T. Xu, Range of user-equilibrium route flow with applications. *Procedia. Soc. Behav. Sci.* **138**, 86–96 (2014)
101. G.L. Nemhauser, L.A. Wolsey, *Integer and Combinatorial Optimization* (Wiley, Hoboken, 1999)
102. K. G. Murty, *Network Programming* (Prentice Hall, Upper Saddle River, 1992)

Chapter 8

An Introduction to Many-Objective Evolutionary Optimization



Dani Irawan  and Boris Naujoks

Abstract This chapter describes the differences between single-objective, multi-objective, and many-objective optimization problems. In multi- and many-objective optimization, often the objectives are conflicting; hence there is no single best point, and a trade-off between the objectives must be considered. Many-objective optimization problems can be more difficult than multi-objective problems mainly because of the curse of dimensionality and because it is also difficult to visualize the trade-off between the objectives. To solve many-objective optimization problems, some algorithms are designed with the challenges in consideration. These algorithms are also described in this chapter, including surrogate-assisted algorithms. Furthermore, several benchmark problems to test and compare the algorithms are discussed.

Keywords Many-objective optimization · Evolutionary algorithms · Benchmarking · Surrogate model · High-dimension visualization

8.1 Introduction

Optimization is the process to bring some things (referred to as objectives) to its best state, i.e., maximum or minimum [2]. Mankind has been optimizing since antiquity. The oldest known record of optimization dates back to 300 BC on works made by Euclid [31].

In this chapter we will consider minimization problems. A maximization problem can be transformed into minimization simply by taking its negative. Often the system is limited by some conditions, known as constraints. The general form of a regular optimization problem is

D. Irawan (✉) · B. Naujoks
Institute for Data Science, Engineering, and Analytics, TH Köln, Köln, Germany
e-mail: irawan_dani@yahoo.com

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} && f : \mathcal{X} \subset \mathbb{R}^n \rightarrow \mathcal{Y} \subset \mathbb{R}, \\
& \text{subject to} && g(\mathbf{x}) \leq 0 && \forall \mathbf{x} \in \mathcal{X} \\
& && h(\mathbf{x}) = 0 && \forall \mathbf{x} \in \mathcal{X}
\end{aligned} \tag{8.1}$$

Here, \mathbf{x} is a vector (with size n) of decision variables inside the decision space \mathcal{X} . The objective function f maps \mathbf{x} into the objective space \mathcal{Y} . The two functions $g(\mathbf{x})$ and $h(\mathbf{x})$ are constraints, known as inequality constraint and equality constraint, respectively.

8.1.1 From Single- to Many-Objective Optimization

In single-objective problems, as the name suggests, only one objective needs to be optimized. When dealing with more objectives, we move on to multi- and many-objective problems. In formal notation, we make a slight change to Eq. (8.1):

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{minimize}} && F : \mathcal{X} \subset \mathbb{R}^n \rightarrow \mathcal{Y} \subset \mathbb{R}^m, \quad F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\
& \text{subject to} && g(\mathbf{x}) \leq 0 && \forall \mathbf{x} \in \mathcal{X} \\
& && h(\mathbf{x}) = 0 && \forall \mathbf{x} \in \mathcal{X}
\end{aligned} \tag{8.2}$$

So, instead of a scalar objective value, we have a vector of it (of size m).

If the problem has two or three objectives ($1 < m < 4$), it is then referred to as a multi-objective problem; if it has four or more objectives ($m \geq 4$), then it is a many-objective problem [21]. As the number of objectives increases, so are the challenges on solving it [7]. Methods applicable on multi-objective problems are anticipated to have difficulties in many-objective cases [43].

8.1.2 Optimality in Multi- and Many-Objective Optimization

When dealing with several objectives, defining ‘‘optimality’’ is a bit different and more complicated. In some cases, when one objective is optimized, the other objectives are also optimized, but, generally, this does not happen. Most often, increasing the quality of one objective will deteriorate one or several other objectives [18]. To compare if a solution is better than other solutions, **Pareto dominance** is defined. A solution \mathbf{x}^* dominates another solution \mathbf{x} :

$$\begin{aligned}
\mathbf{x}^* <_p \mathbf{x} & \iff \forall i : f_i(\mathbf{x}^*) \leq f_i(\mathbf{x}), \quad i = 1, \dots, m \\
& \exists j : f_j(\mathbf{x}^*) < f_j(\mathbf{x}), \quad j = 1, \dots, m
\end{aligned} \tag{8.3}$$

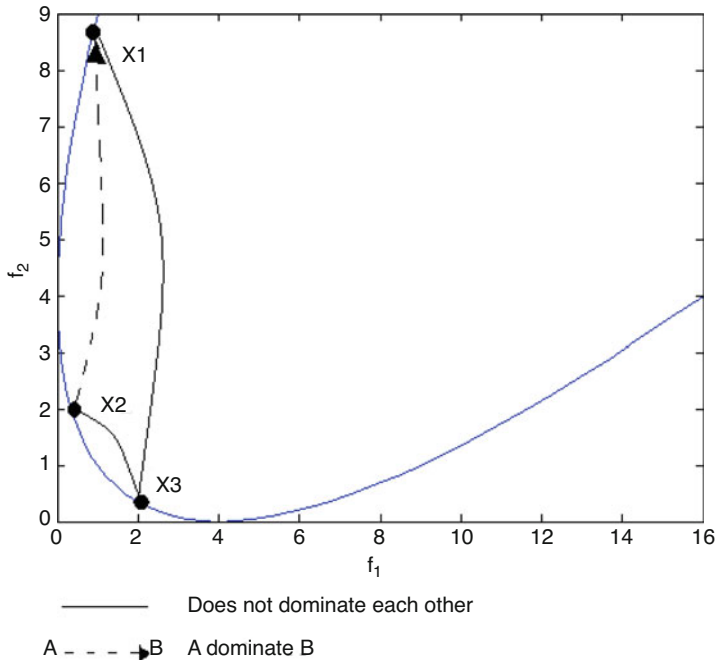


Fig. 8.1 Illustration of Pareto dominance in two-dimensional objective space

An illustration of Pareto dominance relation is presented in Fig. 8.1. In Fig. 8.1, points x_1 and x_3 do not dominate each other, and neither are x_2 and x_3 because the first requirement in Eq. (8.3) is not fulfilled. However, x_1 is dominated by x_2 because $f_1(x_2) \leq f_1(x_1)$ and $f_2(x_2) \leq f_2(x_1)$; thus all requirements are fulfilled.

In multi- and many-objective problems, we are concerned with Pareto optimality: points in the design space where the improvement of one of its corresponding objective values can only be achieved by worsening at least another objective [33]. In formal notation, this means that point \mathbf{x}^* is **Pareto optimal** if and only if

$$\nexists \mathbf{z} \in \mathcal{X} : \mathbf{z} <_p \mathbf{x}^* \tag{8.4}$$

All such \mathbf{x}^* form the **Pareto set**, and their map in the objective space is the **Pareto front**. An example of a Pareto front is presented in Fig. 8.2.

8.2 Evolutionary Algorithm

Evolutionary computation is a field which uses various aspects of biological evolution in computation. Techniques for evolutionary computation date back to

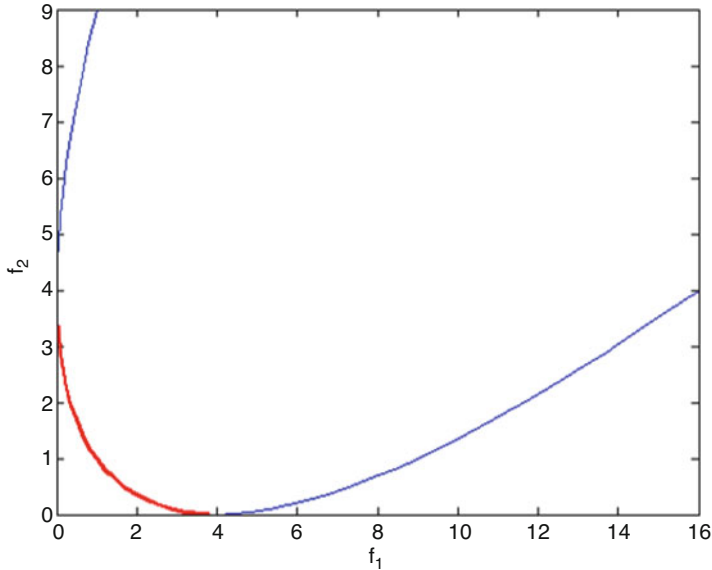


Fig. 8.2 Example of a Pareto front. The figure shows a two-dimensional objective space. The objective values lie on the blue line, and the corresponding Pareto front is highlighted in red

the 1950s [22]. These techniques have been used to study biological processes, arts, and music and to solve some complex engineering problems [4]. We will focus on the application of evolutionary computation for engineering problems.

In nature, organisms attempt to keep, change, or adapt their attributes or characteristics to increase their survivability. This process is optimization in a sense: finding the best configuration of attributes to achieve the best way to survive. Engineers attempted to mimic the evolution process into algorithms for optimization problems in general: replacing “organism attributes” with “free variables” and “survivability” with a more general “fitness function.” These algorithms are called **evolutionary algorithms** (EAs). This set of algorithms is a subset of evolutionary computation [42].

8.2.1 Base Algorithm

Evolutionary algorithms are population-based, meaning that they always generate a set of solutions or design points with their respective objective values, and the best (optimum) solutions are picked from the set. EAs follow a common, general algorithm (Algorithm 1) [5].

In Algorithm 1, t is the generation/iteration counter, $P(t)$ is the population at generation t , $P'(t)$ is the offspring after some variation operator on $P(t)$, and λ is

Algorithm 1 Evolutionary algorithm

```

 $t = 0$ 
 $P(t) \leftarrow$  Initial population of size  $\mu$ 
Evaluate  $P(t)$ 
while Stopping criteria not fulfilled do
  while  $|P'(t)| < \lambda$  do
     $P'(t) \leftarrow$  variation  $P(t)$ 
  Evaluate  $P'(t)$ 
   $P(t+1) \leftarrow$  selection from  $Q \cup P'(t)$ 
 $t = t + 1$ 

```

the intended offspring count. Q is either the empty set or the set of parents that might be considered for selection (more on this in Sect. 8.2.4).

So the algorithm can be read as follows: starting with an initial population, do variation to create offspring, evaluate the whole combined population, select a number of individuals to keep for the next generation, and repeat until a stopping criterion is fulfilled. This is the basic algorithm; however, there are variations on the implementation as it will be discussed further in this chapter.

The solutions/design points in EA are called **individuals**, and a set of individuals form a **population**. Each individual is represented by a sequence of **genes** which form a **chromosome**. The chromosome **encodes** or **represents** the variable values. **Encoding** is how the variables are represented in the EA [44]. The most common encoding is either binary (all variables are represented by only ones or zeros) or real-valued.

In Algorithm 1, the genetic operator that will improve solutions is variation. Usually this is done in the form of **recombination** or **mutation** [5]. Recombination is the mixing of chromosomes from several different individuals (called **parents**) through a selection procedure to create **offspring**. Mutation is the process of randomly changing the chromosome information within an individual.

The last step of the algorithm is **selection**. This step is done by keeping good individuals based on some performance metrics and using them for further iterations while the rest of the population is discarded. This step is used to keep the population at a manageable size and to foster progress.

8.2.2 Recombination

Recombination is one of the operators used to modify the individuals in EA. Recombination is intended to combine characteristics of several individuals (parents) to produce offspring with new, different characteristics. Note that the number of parents can be more than two [5].

In EA, recombination is done simply by taking the chromosome of at least two parents and using these values to create new individuals with different chromosomes. The procedure can be performed using several possible methods. The most

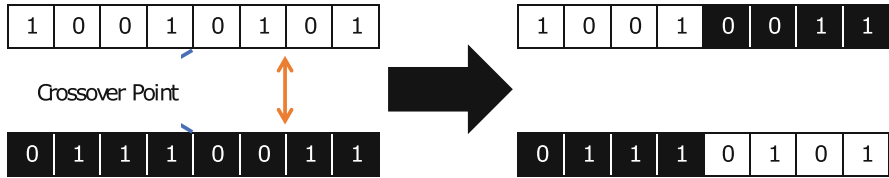


Fig. 8.3 Example of recombination: the single point crossover. Part of the chromosomes from two parents (left) are switched resulting in two offspring (right) with different chromosome sequences

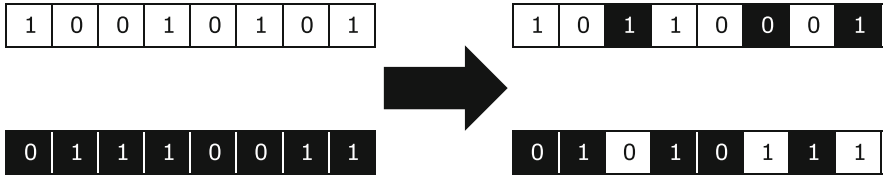


Fig. 8.4 Recombination using uniform crossover. The crossover can happen anywhere within the chromosome

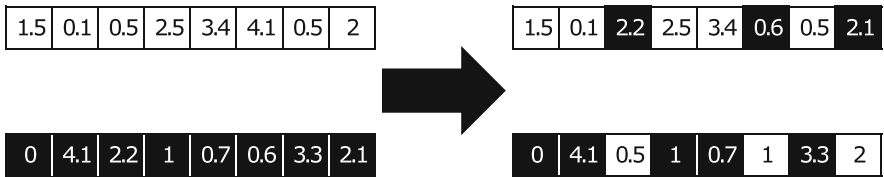


Fig. 8.5 Recombination using uniform crossover on real values. Similar with the binary counterpart, but instead of the bits, each block is a variable by itself

common one is by swapping gene values. More generally, some transformation functions are applied to gene values, like in the case in which the new recombining gene is obtained taking the average value of the corresponding genes belonging to the parents. One of the most widely used recombination operator is the simulated binary crossover (SBX) (see [1]).

The simplest recombination operation is single point crossover in binary encoding. In **single point crossover**, starting from a crossover point, the bit value of two parent chromosomes is swapped (see Fig. 8.3). In the bit swapping shown in Fig. 8.3, the offspring genes before the crossover point are unchanged, while after the crossover point, the genes are swapped. Another variant is **uniform crossover** [5] where the crossover is triggered for each bit in the chromosome as shown in Fig. 8.4.

The recombination operator is also available in real encoding. In principle, the requirement for a crossover operator is that the offspring are a combination of their parents. An example of real-valued crossover is the **discrete crossover** [5] where the variable values are swapped, similar with the binary chromosome, but instead of bits, the real values are swapped, illustrated in Fig. 8.5.



Fig. 8.6 Example of bit switching mutation. A random point on the chromosome of an individual value is changed

The recombination operation is intended to explore the search space locally around the parents' population. Even though the offspring are different from their parents, they would still retain some degree of similarity with their parents.

8.2.3 Mutation

Mutation is an operator intended to keep the diversity of the population and prevent the population from gathering in a local optimum.

Usually, mutation is programmed to happen randomly within the population, following a probability distribution. If mutation happens, the individual's chromosome is modified. In binary encoding EA, this can easily be done by bit switching of the genes (see Fig. 8.6). In real-valued EA however, things get more complicated.

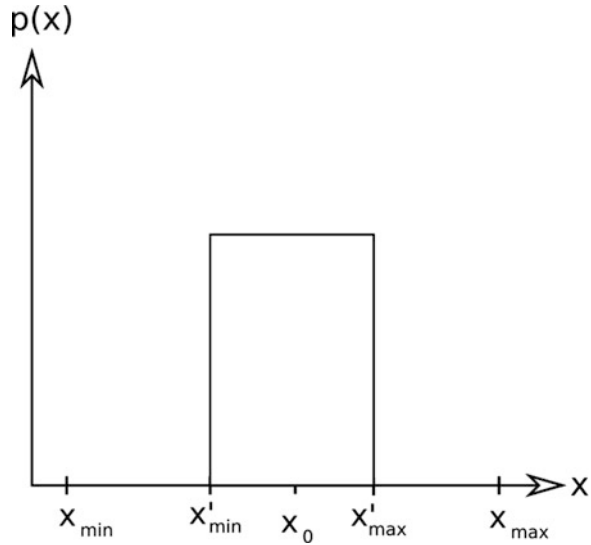
In binary-encoded EAs, each gene has only two possible values; hence, the mutation will change the gene value to its complement, i.e., zero to one or vice versa. In real-valued EA, unlike in binary-encoded one, there are many possible values the gene can have after the mutation. Mutation in real-valued EA is done by changing the genes' values to other real numbers. The new values can be any number. However, because each real-valued gene encodes more information than a binary-valued one, usually the changes allowed to the chromosome are limited and known as **creep mutation** [44]. The creep mutation allows the mutated values to follow some distribution around the original values. Some commonly used distributions are the uniform (shown in Fig. 8.7), Gaussian [15, 44], or polynomial distribution [10].

8.2.4 Selection

Recombination and mutation introduce new individuals to the population. With each addition, the population size grows; and when implemented on a computer program, this means more memory consumption. Nowadays, however, with the growth of computation technology and memory capacity, people are less concerned with memory consumption.

Another problem with keeping all individuals is the probability of regressing. Older population members are supposed to have worse qualities than the new

Fig. 8.7 Example of the creep mutation with uniform distribution. The initial point, i.e., the parent, is located at x_0 ; after creep mutation it could take any point between x'_{min} and x'_{max} with equal probability. Without creep mutation, the probability density is spread over x_{min} and x_{max} instead



generations. If the old individuals are always kept, there are chances that the population will return to an older state through recombination, thus regressing instead of progressing to a better state.

The solution to these problems is to truncate (cut off) the population. However, a new question arises, “which individuals should be kept and which should be removed?” The answer to this question is defined in the selection operator. Selection operator will foster progress by removing individuals which are considered to be bad and will increase the chance to create better offspring.

We may want to keep all the “best” individuals, but, actually, selection operators should not always favor the most fit individuals as it would easily lead to stagnation [15, 44]. So, basically, the selection operator has multiple purposes: prevent stagnation, foster progress, and avoid making the population too large. Also note that the population does not have to be at a constant size, some algorithms do use a varying or adaptive population size such as the GAVaPS [3], or the growing SMS-EMOA [23].

The population on which the selection is conducted is determined by the selection scheme. There are two selection schemes: the plus and comma schemes. The plus scheme is where both the parents and offspring are considered to be kept, while the comma scheme disregards the parents [15, 17], i.e., the parents are always discarded. The schemes are usually written as $(\mu + \lambda)$ and (μ, λ) for the plus and comma schemes, respectively, where μ is the number of parents and λ is the number of offspring.

After choosing the selection scheme, then the rule on how to select the individuals need to be decided (known as the selection mechanism [15, 17]). Examples of selection mechanism are tournament selection [44], fitness-proportional selection [5, 44], and non-dominated sorting [12].

The choice of the scheme and mechanism usually differentiates the EAs. For example: SMS-EMOA uses $(\mu + 1)$ scheme with non-dominated sorting and S-metric selection, NSGA-II uses $(\mu + \mu)$ scheme with non-dominated sorting and crowd-distance selection, and NSGA-III uses $(\mu + \mu)$ non-dominated sorting and reference-point distance. These operators will be discussed further in Sect. 8.3.3.

8.3 Multi-Objective Optimization

This section will discuss how to solve multi-objective optimization problems. Several methods as well as some performance metrics to compare solutions will be described.

8.3.1 Method Classifications Based on Preference-Imposing Timing

In Sect. 8.1.2, it was mentioned that in multi- and many-objective problems, we are concerned with the solutions in the Pareto set. This would imply that in a decision making process, decision makers must choose the “best” design from the Pareto set considering his/her preference on the trade-off between the objectives (the Pareto front). The preference can be imposed before (*a priori*), after (*a posteriori*), or progressively within the optimization loop.

8.3.1.1 A Priori Method

A priori methods simplify the problem by transforming the problems into one or a series of single-objective optimization problems (SOP). Several methods that fall into this category are described below.

Lexicographic Method

The lexicographic method considers an absolute importance order [16]. The method is similar with the process of sorting words in dictionaries [28]:

- Sort by the first letter
- For the same first letter, then sort by the second letter
- Continue to the next letters until all items have different ranks or all letters in the word are used

Analogously, in optimization, the method is the same; however, instead of letters, we have objectives, and the number of objectives is always the same for all designs. The objectives must be ordered by importance.

Aggregation Method

Another way to deal with the ambiguity of optimality in many-objective problems is by summing up the objective values, thus transforming the problem into a single-objective problem. Preference is imposed by weight factors, i.e.

$$\underset{\mathbf{x}}{\text{minimize}} \quad F(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{x})w_i, \quad (8.5)$$

Usually the weight w should sum up to 1:

$$\sum_{i=1}^m w_i = 1, \quad (8.6)$$

After the aggregation of the objective functions, it is then only a matter of solving single-objective optimization problems. If the method is run in series with different weights, it will form the Pareto front and set. However, this method has a major weakness that it cannot find points on the concave regions of the Pareto front [34, 46].

Note that when the objectives have different orders (e.g., one objective in hundreds, the other in millions), assigning weights would be difficult because objectives with a higher order would be considered very important compared to objectives with lower order. When this happens, a normalization factor which transforms the objectives into similar order and range should be used.

8.3.1.2 *A Posteriori* Methods

In *a posteriori* methods the decision makers will be given a set of solutions (the Pareto set) and their corresponding objective values (the Pareto front). The decision makers can then choose their preferred designs from the given solutions.

With regard to Sect. 8.2.1, it was mentioned that EAs are population-based methods. Population-based method will have several candidate solutions, each represented by an individual. Using appropriate genetic operators, the candidate solutions can be guided to find different trade-offs in the objective space. This means, from a single optimization loop, instead of obtaining a single solution, the Pareto front could be approximated. It is then interesting to use EAs to solve multi- and many-objective optimization problems. Some EAs are described in Sects. 8.3.3 and 8.4.2.

8.3.1.3 Progressive Methods

In the two sections above, we have mentioned that the decision makers can input their preferences before or after the optimization loop. The other possibility is to input their preferences *during* the optimization. The decision making and optimization are intertwined, i.e., within the optimization loop, the decision makers need to give preference information [41].

One way to do this is by generating a set of solutions and requiring the decision makers to pick their most favorite. These favorite solutions are taken to update the preference information, and then new solutions are generated. The process could be repeated until a stopping criterion is reached.

8.3.2 Solution Quality Assessment

Comparing solutions in multi- and many-objective problems is not a trivial task because what we have is a set of non-dominated solutions instead of only one solution. This would imply that we need to define what makes a non-dominated set better than another non-dominated set. These measurements are called performance metrics or performance indices.

Performance metrics are usually based on three criteria: cardinality, accuracy, and diversity [35, 36]. **Cardinality** simply means the number of points in the non-dominated set; **accuracy** measures convergence to the real Pareto front; and **diversity** measures how well spread the solutions are in the objective space. A performance metric can measure more than one criteria simultaneously.

The number of performance metrics currently available is vast. This section will only introduce the top two most used metrics between 2005 and 2013: hypervolume and generational distance. Some other metrics are also described in Sects. 8.3.3 and 8.4.2; the metrics in the section are used to rank solutions within the population; e.g., the crowding distance, non-dominated ranking, etc. Other popular metrics are the ϵ -indicator [48] and R-metric [24] which can compare performances of a pair of solution sets in all three aforementioned criteria simultaneously [36].

8.3.2.1 Hypervolume

The hypervolume is the most widely used performance metric [36]. Hypervolume is a generalization of the area (2D), or volume (3D) in higher dimensions. In a biobjective problem, the hypervolume is measured as the area covered by the non-dominated set with respect to a reference point (see Fig. 8.8).

The maximum hypervolume can only be achieved by the real (possibly continuous) Pareto front, thus maximizing hypervolume is a straightforward and general goal to approach the real Pareto front [43]. The hypervolume can also measure

Fig. 8.8 Example of hypervolume measure in a 2D minimization case with a reference point in the top right (large f_1 and f_2). The hypervolume (yellow) is then formed by connecting lines to the non-dominated set (green dots)

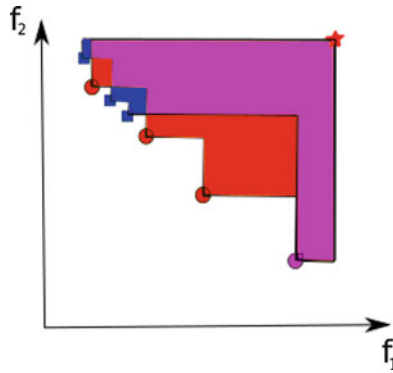
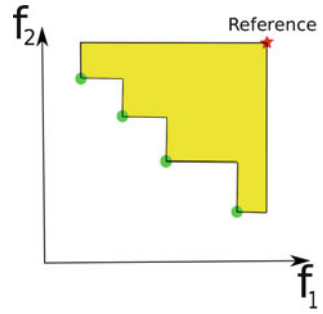


Fig. 8.9 How diversity affects hypervolume. A well-spread non-dominated set (red circles), while the other set (blue rectangles) has a cluster of solutions placed in the area with small f_1 . The red areas are the areas dominated only by the circles; the blue areas are dominated only by the rectangles; and the purple area is dominated by both sets. Larger hypervolume can be achieved when the non-dominated set is well spread

diversity because different distributions of non-dominated sets will give different hypervolume measurements (see Fig. 8.9).

8.3.2.2 Generational Distance

Generational distance (GD) [8, 40] is the second most used performance metric. To measure generational distance, the real Pareto front must be known. This requirement limits the use of GD to be used only on test problems; it is not applicable to general problems where the real Pareto front is unknown.

However, when new methods are proposed, traditionally, the methods are compared against previously known methods on benchmarking test functions (see Sect. 8.6). In these cases, GD can serve as a performance metric to compare the methods.

Fig. 8.10 GD is calculated as distance from each non-dominated points (blue star) to the closest reference point (red circle)

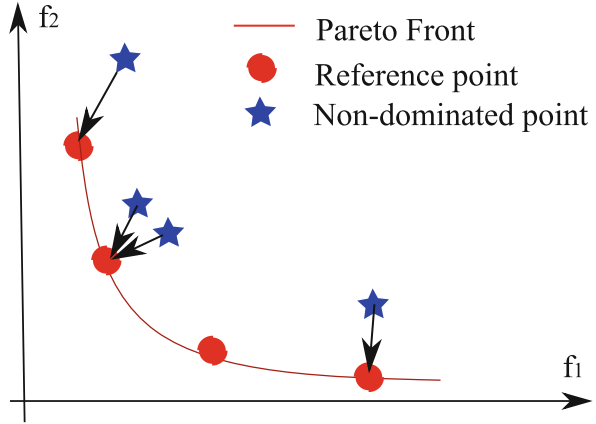
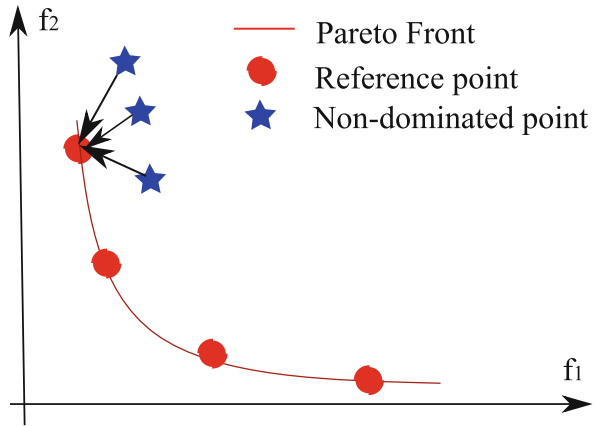


Fig. 8.11 The non-dominated points are not well spread, but the GD measure is good (small value)

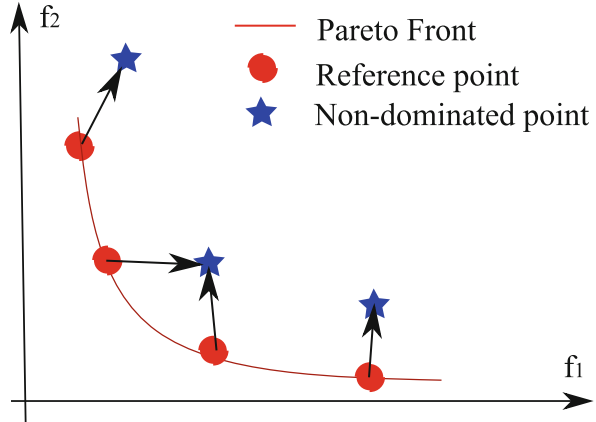


GD is calculated as the average (usually Euclidean) distance of all non-dominated points to its closest reference point. Regularly, the reference points are spread all over the real Pareto front (see Fig. 8.10).

It should be noted that the reference points in GD serve different purposes compared to the reference point in hypervolume measurement. In hypervolume measurement, the reference point has quite a bad quality in terms of convergence to the real Pareto front; hence, it is only used as a limit of how far away the edges of the hypervolume are. In GD, the reference points are actually target points, placed on the Pareto front.

A small GD implies that all non-dominated points (i.e., the map of our best solutions in the objective space) are located near the real Pareto front which is what we want to achieve. However, a small GD does not imply the non-dominated points are well spread because it could be that all the points are gathered (converged) around a single reference point (see Fig. 8.11).

Fig. 8.12 IGD is the inverse of GD, calculated as distance from each reference point (red circle) to the closest non-dominated points (blue star)



To measure diversity it is better to use the inverse GD (IGD) [8]: instead of calculating the average distance of the solutions to its nearest reference point, we calculate the average distance from all reference points to their closest non-dominated point (Fig. 8.12). Similar to GD, a small IGD is desirable. However, this is a completely different measure because if all non-dominated points converge to a single reference point, other reference points will have a large minimum distance to non-dominated point. Hence, the GD would be small, but the IGD will be large.

8.3.3 Algorithms Designed for Multi-Objective Optimization Problems

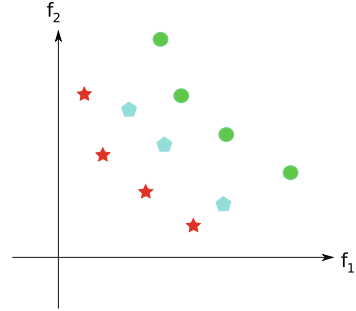
Some EAs are actually designed to tackle the problem of finding the Pareto front of multi-objective problems. These algorithms are called EMOAs (evolutionary multi-objective optimization algorithms) or MOEAs (multi-objective evolutionary algorithms).

8.3.3.1 NSGA-II

NSGA-II is the **elitist non-dominated sorting genetic algorithm** by Deb et al. [12]. Currently it is well-known and the most frequently used EMOA. In NSGA-II, broadly speaking, any recombination and mutation method can be used. The defining feature of NSGA-II is its selection methods. The primary selection method is called **non-dominated sorting**:

1. The non-dominated front is ranked as the first front.
2. Remove the first front; the second front is the non-dominated individuals when the first front is removed.

Fig. 8.13 Illustration of non-dominated sorting in a 2D objective space. The stars, pentagons, and circles are members of the first, second, and third fronts, respectively



3. Remove the first and second front; the third front is the non-dominated individuals when the first and second front are removed.
4. Continue removing and ranking until all points are ranked.

The ranking goes on until all individuals assigned a rank. An example is shown in Fig. 8.13.

Starting from a population/parent P with size μ , a set of offspring P' with same size μ is created. Non-dominated sorting is then applied to the combined $P \cup P'$. After assigning ranks, individuals that will be used on the next generation are selected. The member of each ranked front is counted, and these counts are summed from the first front to lower ranks until it is equal or exceeds μ , and we call the final sum result as K . All other fronts will not be used as parents for the next generations; hence, they are discarded. If the sum of the counts K is equal to μ , no further selection is required, and we have the parents for the next generation. However, if K is larger than μ , i.e., we still have too many population member, a further, secondary selection is conducted.

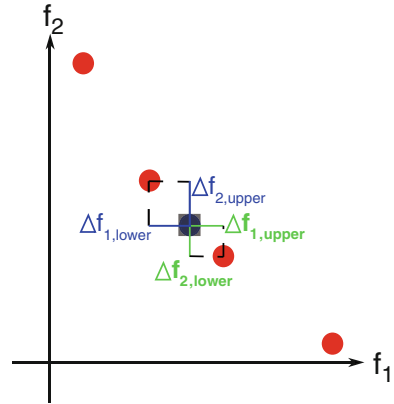
NSGA-II uses a secondary selection called crowding distance. The crowding distance is calculated as the sum of distances to the next higher and lower values in each dimension (Fig. 8.14). The individuals with the smallest distances to its neighbors will be removed. The overall runtime for NSGA-II is $O(\mu \log^{d-1} \mu)$ per generation [43]; we can see that the number of dimension d causes an exponential increase for the runtime.

As a summary, the NSGA-II algorithm is shown in Algorithm 2. Notice that the NSGA-II algorithm follows the base algorithm shown in Sect. 8.2, only expanding the selection procedure right after $P'(t)$ is evaluated. R_i in the algorithm are the individuals with non-dominated sorting rank i .

8.3.3.2 SMS-EMOA

SMS-EMOA stands for **S-metric selection EMOA** by Emmerich et al. [19]; the goal is to maximize the S-metric value of the population. **S-metric** is simply the hypervolume.

Fig. 8.14 Illustration of the crowding distance in a 2D objective space. The light blue point will be removed because its distances to the neighboring points are the smallest



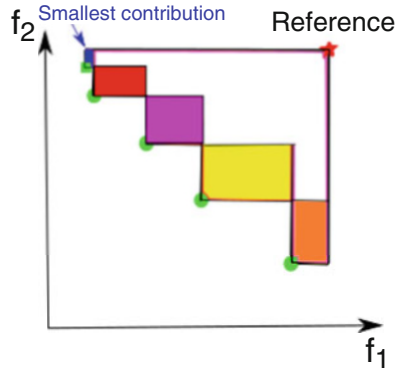
Algorithm 2 NSGA-II

```

t = 0
P(t) ← Initial population of size μ
Evaluate P(t)
while Stopping criteria not fulfilled do
    while |P'(t)| < μ do
        P'(t) ← variation P(t)
    Evaluate P'(t)
    Non-dominated sorting on P(t) ∪ P'(t)
    i ← 1
    K ← 0
    while K < μ do
        P(t + 1) ← Ri
        i ← i + 1
        K ← K + |Ri|
    if K > μ then
        Crowding distance selection on P(t + 1)
    t = t + 1
    
```

SMS-EMOA uses a **steady-state selection** scheme, meaning that only 1 new individual is produced from the mating procedure and from the parents and 1 offspring, 1 point is removed. The parents selection for mating is equiprobable. As the primary selection operator, SMS-EMOA uses non-dominated sorting as used in NSGA-II. The worst ranked front from non-dominated sorting can still have several points in it; thus a secondary selection is conducted: the removed point should have the smallest contribution on the worst-front hypervolume. Figure 8.15 depicts a front where secondary selection is conducted. The top-left point is the smallest contributor to the total hypervolume; thus it will be removed from the population. Note that the contribution of the edge points can be larger or smaller depending on the reference point location; hence different reference points could lead to different selections.

Fig. 8.15 Illustration of SMS-EMOA selection procedure in a 2D objective space. Each box is the contribution of a single point to the total hypervolume



Algorithm 3 SMS-EMOA

```

t = 0
P(t) ← Initial population of size μ
Evaluate P(t)
while Stopping criteria not fulfilled do
    while |P'(t)| < 1 do
        P'(t) ← variation P(t)
    Evaluate P'(t)
    Non-dominated sorting on P(t) ∪ P'(t)
    i ← 1
    K ← 0
    while K < μ do
        P(t + 1) ← Ri
        i ← i + 1
        K ← K + |Ri|
    if K > μ then
        Smallest contributor removal from P(t + 1)
    t = t + 1
    
```

The hypervolume contribution of a point \mathbf{x} is defined as the hypervolume loss when \mathbf{x} is removed from the non-dominated set. The simplest method to measure it is by comparing the hypervolume size before and after removing the point. With $\mu + 1$ number of points, $\mu + 2$ hypervolume calculation must be conducted (once for the whole non-dominated set and $\mu + 1$ times for removal of each point) using this algorithm. The runtime of a generation of SMS-EMOA is $O(\mu^{\frac{d}{2}+1})$ [43]; the d also is the exponential factor for the runtime.

The algorithm is shown in Algorithm 3 [19]. Again, it follows the base algorithm. It is even similar to NSGA-II, with changes on the size of $P'(t)$, which is only one in SMS-EMOA, and on the secondary selection.

8.4 Many-Objective Optimization

8.4.1 *Challenges in Many-Objective Optimization*

First, recall that in many-objective optimization problems, at least four objective functions are considered (Sect. 8.1). There are several complications when dealing with many-objective problems. First, methods applicable on single-objective problems can only be used if the original many-objective problem is transformed into a single-objective by using aggregation methods mentioned before (see Sect. 8.3.1.1). Furthermore, even methods applicable on multi-objective problems may have bad performance or are even not feasible to use in many-objective cases. The reason for the latter problem usually stems from the so-called curse of dimensionality. In other cases it is due to the problems being too expensive to evaluate.

8.4.1.1 Curse of Dimensionality

Curse of dimensionality is a term which expresses how an increased dimension leads to an extremely high increase of cost or deterioration of solution quality. In the previous section, it has been shown that the number of dimensions d is the exponential factor for runtime. Hence increasing d leads to an exponential increase in the runtime.

Curse of dimensionality can be mitigated by reducing the problem size. The methods to reduce the size can simply be done by ignoring some objectives (similar to what was done in lexicographic method mentioned in Sect. 8.3.1.1) or using model reduction methods such as principal component analysis (PCA) [39].

Another problem with dimensionality in many-objective optimization is the loss of pressure to find the Pareto front. With increasing dimensionality, non-dominatedness is easier to achieve [29]. With low pressure, the population will converge to the Pareto front only slowly.

8.4.1.2 Expensive Evaluation

Expensive evaluation means that the evaluations have very high cost. However, “cost” here is not limited to financial cost. Sometimes, it is indeed the monetary cost that is expensive, but it can also be other kinds of costs. The cost can be evaluation time, manpower or computational power required, etc. For example, when the function evaluated is actually a result of a simulation, each of the simulation can take several minutes, hours, or even days. If a single simulation takes a long time to finish, the whole optimization process—which requires several simulations to be run—will also consume a lot of time to finish. Another example is when to evaluate a design, a prototype must be manufactured. This implies costs in both time and

money. In these cases it is impossible or inefficient to evaluate all required points to find the Pareto front.

Expensive evaluation is not an exclusive problem which is only applicable to many-objective optimization. Even a single-objective optimization task would consider expensive evaluation as a challenge. A popular solution is using surrogate model/function which is applicable on single-, multi-, and many-objective optimization problems. The surrogate model is a prediction and simplification of the real model, based on some early samples. Further explanation of surrogate models is available in the other chapters, while a specific application of surrogate modelling in multi- and many-objective optimization is available in Sect. 8.5.

8.4.1.3 Visualization Challenge

When the number of objectives exceeds 3, visualization becomes a problem. The regular visualization using scatterplot is limited to view a three-dimensional system projected into a plane (two dimensions). Some recent researches attempt to take it further to view four-dimensional systems by using a projection onto a three-dimensional space [6]; however, it does not change the fact that visualization will be limited. Some methods to visualize the objective space in a higher dimension are presented in Sect. 8.4.3.3.

8.4.2 Algorithms Designed for Many-Objective Optimization Problems

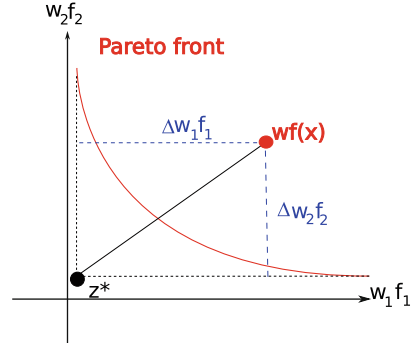
Due to the challenges posed by many-objective problems, MOEAs for multi-objective problems are difficult to use. Researchers around the world devise new EAs specifically designed for many-objective problems. The algorithms are called MOEAs or EMOAs.

8.4.2.1 MOEA/D

MOEA/D is the **MOEA based on decomposition** proposed by Zhang and Li [46]. **Decomposition** means transforming the many-objective problem into finite number μ of SOPs by using some aggregation methods. Zhang and Li use the weighted sum and the Tchebycheff method to decompose the many-objective problem. The Tchebycheff method (see Fig. 8.16) has the advantage of being able to find points in the non-convex region of the Pareto front [46].

Each subproblem will lead to a single point in the Pareto front, so in the end μ solutions will be obtained. The algorithm limits mating only between T number of neighbors with equal probabilities (equiprobable selection). Diversity is not

Fig. 8.16 Illustration of the Tchebycheff decomposition method for MOEA/D in a two-dimensional weighted objective space. The Tchebycheff method aims to minimize the maximum weighted distance $\Delta w_i f_i$ of all objectives i with respect to the ideal point z^* , therefore pulling the solution closer to the Pareto front



“preserved” in the run, rather it is set at the beginning by choosing different weight vectors for aggregation in each of the μ subproblems.

For the final result, MOEA/D maintains an external population, i.e., all the best results are kept separately. Every time a better solution is found, the external population is updated. In terms of non-dominated sorting, all evaluated solutions are collected, and only the individuals ranked 1 from this large set are kept.

Compared to NSGA-II, the relative expected runtime for MOEA/D has a factor of $O(T)/O(\mu)$ per generation [46]. Using the NSGA-II runtime in Sect. 8.3.3, we can determine that the overall runtime per generation is $O(T \log^{d-1} \mu)$. Generally T is lower than μ ; hence it is faster than NSGA-II and was expected to be applicable on many-objective problems.

The algorithm is presented in Algorithm 4. In the algorithm, z is the ideal point, taking the best values attained in each objective. EP is the aforementioned external population. Also, instead of dealing with population $P(t)$, MOEA/D considers each individual $P_i(t)$ separately.

Algorithm 4 MOEA/D

```

 $t = 0$ 
EP =
for  $i = 1, \dots, \mu$  do
  Define neighbour  $B_i$  of size  $T$ 
   $P_i(t) \leftarrow$  Initial individual
  Assign weight vector  $w_i$  for individual  $P_i(t)$ 
  Evaluate  $P_i(t)$ 
   $Q_i(t) \leftarrow P_i(t) \times w_i$ 
Initialize  $z$ 
while Stopping criteria not fulfilled do
   $P'_i(t) \leftarrow$  variations from  $B_i$ 
  Evaluate  $P'_i(t)$ 
  Update  $z$ 
  Update Neighbour Solutions using  $P'_i(t) \times w_j, j \in B_i$ 
  Update EP

```

8.4.2.2 NSGA-III

NSGA-III by Deb et al. [11] is a recent addition to the library of MOEA algorithms. As the name suggests, it has a similarity with NSGA-II. In fact, NSGA-II is also invented by Deb and his colleagues [12].

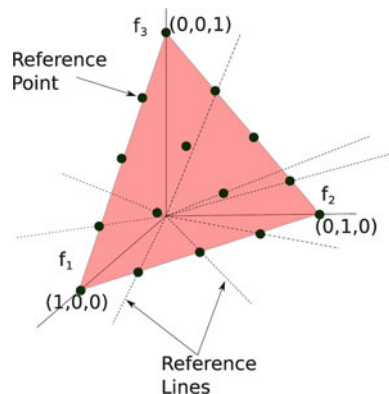
As a note, originally, NSGA-III can work well only on many-objective problems. However, Seada [37] together with Deb modified the algorithm to work on lower dimensions in their subsequent publication on unified NSGA-III (UNSGA-III). This section, however, only describes the original NSGA-III.

NSGA-III starts with recombination and mutation to generate offspring P' , then uses the same non-dominated sorting as in NSGA-II and SMS-EMOA, but that is where the similarity ends. For the secondary selection operator, NSGA-III is more akin to MOEA/D: it uses reference points. While in MOEA/D we use one reference point with multiple search directions based on the weight vectors (i.e., for each weight, one optimization loop is conducted), NSGA-III opts to use multiple reference points representing different weights of the objectives (i.e., in a single optimization loop, all weights are considered).

The number and locations of reference points can vary depending on preference, Deb and Jain [11] showed that the method can work with both structured and unstructured reference points. For the structured reference points, Deb and Jain place the reference points on a normalized hyperplane which is equally inclined to all objective axes and has an intercept of one on each axis (Fig. 8.17). If we consider an M -objective problem, and each objective-axis is divided into p partitions, the total number of reference points is $\binom{M+p-1}{p}$. Each of the reference point, paired with the ideal point, will create a reference line which becomes the basis for the secondary selection.

In NSGA-III, the number of offspring is set to be the same as the size of the parent population μ , so for the selection we have 2μ individuals to be considered. The difference with NSGA-II starts after the non-dominated sorting when the population count after non-dominated sorting is larger than μ . This is when we need to conduct a secondary selection. The secondary selection in NSGA-III is based on the distance

Fig. 8.17 An example of the structured reference point on a normalized hyperplane used in NSGA-III. A three-dimensional objective space is divided into four regions in each axis giving $\binom{3+4-1}{4} = 15$ points. Each reference point, paired with the normalized origin, forms a reference line creating 15 reference lines



of the last ranked front to a reference line instead of the distances to its neighbor. So instead of comparing distances of $K - \mu$ individuals to all other $2\mu - 1$ points in each dimension, NSGA-III only needs to calculate distances between the $K - \mu$ individuals to $\binom{M+p-1}{p}$ reference lines.

The algorithm is presented in Algorithm 5. As mentioned before, the change from NSGA-II starts in the secondary selection. Instead of crowding distance, the distances to reference points are checked. It may look more complicated due to the frequent distance measurement.

Algorithm 5 NSGA-III

```

 $t = 0$ 
 $P(t) \leftarrow$  Initial population of size  $\mu$ 
define reference lines  $L$  of size  $H$ 
Evaluate  $P(t)$ 
while Stopping criteria not fulfilled do
  while  $|P'(t)| < \mu$  do
     $P'(t) \leftarrow$  variation  $P(t)$ 
    Evaluate  $P'(t)$ 
    Non-dominated sorting on  $P(t) \cup P'(t)$ 
     $i \leftarrow 1$ 
     $K \leftarrow 0$ 
    while  $K + |R_i| \leq \mu$  do
       $P(t+1) \leftarrow R_i$ 
       $i \leftarrow i + 1$ 
       $K \leftarrow K + |R_i|$ 
    Measure distance from  $P(t+1)$  to all  $L$ 
    Associate each individual in  $P(t+1)$  to nearest  $L$ 
     $C_j \leftarrow$  Count of assoc. solutions from  $P(t+1)$  for line  $j, j \in 1, \dots, H$ 
    Measure distance from  $R_i$  to all  $L$ 
    Associate each individual in  $R_i$  to nearest  $L$ 
     $c_j \leftarrow$  Count of assoc. solutions from  $R_i$  for line  $j, j \in 1, \dots, H$ 
    while  $K < N$  do
       $j_{\text{least}} \leftarrow \underset{j \in 1, \dots, H}{\operatorname{argmin}}(C_j)$ 
      if  $c_{j_{\text{least}}} = 0$  then
        Remove line  $L_{j_{\text{least}}}$  from consideration in current  $t$ 
      else
         $A \leftarrow$  nearest member of  $R_i$  to line  $L_{j_{\text{least}}}$ 
         $P(t+1) \leftarrow P(t+1) \cup A$ 
         $R_i \leftarrow R_i \setminus A$ 
         $c_{j_{\text{least}}} \leftarrow c_{j_{\text{least}}} - 1$ 
         $C_{j_{\text{least}}} \leftarrow C_{j_{\text{least}}} + 1$ 
     $t = t + 1$ 

```

8.4.3 High-Dimension Visualization Techniques

To express how difficult it is to visualize a space with dimension higher than 3, let us review how we normally see an image, namely, the scatterplot visualization. The

visualizations, often presented on paper or screen, appears on a 2D media, which also means it is limited to only show anything in 2D. What we normally do in single or multi-objective optimization is to present the objective space on a Cartesian coordinate system. With this method, we can easily view 1D and 2D spaces, and, by using projection methods, 3D space can also be viewed on a 2D plane. The problem is, physically, we cannot construct or observe higher dimensions [45].

In many-objective optimization, the number of objectives is higher than 3. This implies that using the scatterplot method would limit us to only view 3D or lower dimension spaces.

8.4.3.1 Bubble Chart

A bubble chart is an extension of the usual scatterplot. Instead of only using the position of the points to visualize the objective space, a bubble chart also utilizes other properties of the points, e.g., point sizes and colors [38].

By using different point sizes, the fourth dimension can be visualized, i.e., larger points indicate larger fourth axis value. Colors can also be used with the help of a color scale. An example of a bubble chart is shown in Fig. 8.18. In the figure, the

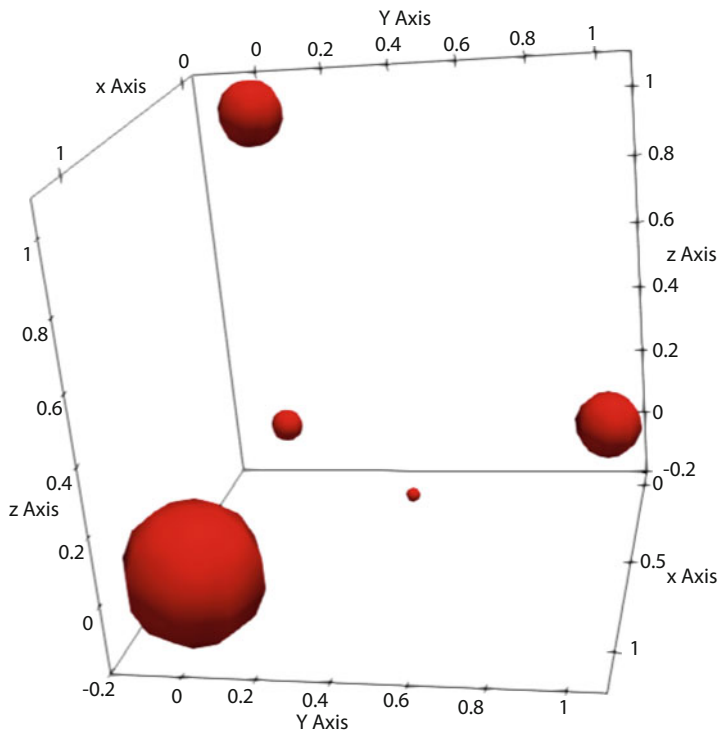
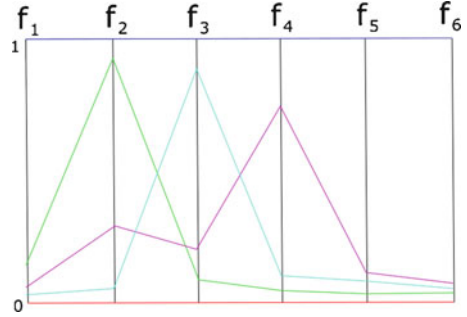


Fig. 8.18 Example of a bubble chart with five points. The bubble chart visualizes a 4D problem

Fig. 8.19 Example of a parallel plot. The figure depicts a six-objective problem with normalized objective values



first three axes are indicated by the x , y , and z axes, while the fourth axis is indicated by the point sizes. This technique has the advantage of being a very easy extension of the scatterplot. It offers all the properties of a scatterplot plus the possibility of visualizing additional dimensions. However, as the fourth and higher dimensions are visualized by the properties of the point, it may be more difficult to discern domination relations, i.e., whether one solution dominates other solutions. Further, when the number of vectors to be displayed is large, the plot will be easily cluttered by points with large sizes.

8.4.3.2 Parallel Plot

In parallel plots, all objective functions are represented by parallel lines. Points in each line represent the magnitude (usually normalized) of their corresponding objective value. A line connecting the points represents the objective value realizations of a solution. An example is shown in Fig. 8.19. In the figure, the horizontal line with all values at zero (red) represents the ideal point; the horizontal line with all values at one (blue) represents the nadir point. Each of the other three lines (green, turquoise, and purple) represents objective value realizations of a solution.

This visualization method is very easy to use and simple to understand. When used to view a non-dominated front, the domination relation and spread can be observed, but not the shape (convex/non-convex, linear, etc.) [38].

The domination relation can be observed as shown in Fig. 8.20. The blue line has a higher value in all objectives compared to the red line, i.e., the red line dominates blue (minimization case). When the objective vector does not dominate each other, the lines will cross each other at least once.

The downside of this visualization method is when the number of vectors to be shown is large, the figure becomes cluttered. It would be difficult to see and trace the lines; as an example, see Fig. 8.21. It is still possible to differentiate the lines in the figure, but imagine if more and more lines are added, the figure would be more difficult to comprehend.

Fig. 8.20 The red line dominates the blue line

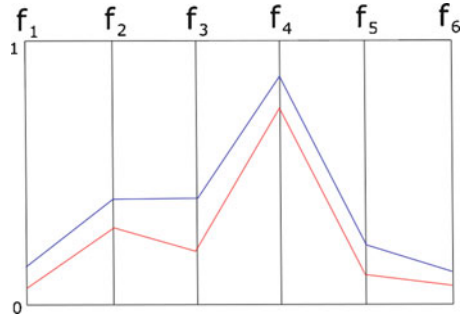
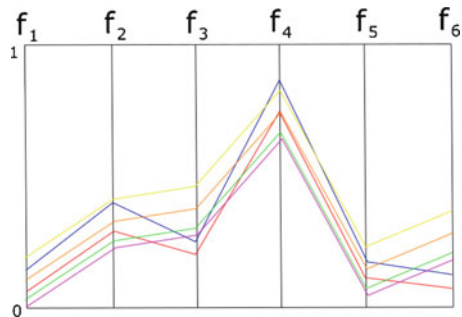


Fig. 8.21 Parallel plot of six similar solutions (not necessarily non-dominated)



8.4.3.3 Glyph Plot

A glyph plot is also known as radar, spider, or star plot. Glyph plots can be considered similar to parallel plots, except that in glyph plots, each axis has different directions as opposed to parallel lines. A glyph plot can be considered as a parallel plot that has been bent and connected. Examples are shown in Figs. 8.22 and 8.23. For minimization case, a solution is dominated when all points of the glyph are farther away from the center compared to other glyphs.

Due to the similarities between glyph and parallel plots, the advantages and disadvantages are also similar. Both plots are simple and easy to use, and they can show domination relations, but they can be easily cluttered. For Pareto front visualization, this technique also cannot distinguish the front shape.

8.5 Surrogate Model in Multi- and Many-Objective Optimization

Let’s recall one of the challenges mentioned in Sect. 8.4.1: expensive evaluation. When the evaluation of the objective functions has a high cost, the number of evaluations must be kept low. In multi- and many-objective problems, this is even

Fig. 8.22 An example of star plot for a six-objective problem. In the plot, three solutions are drawn, each forming a glyph with different stroke styles (solid, dash, dots)

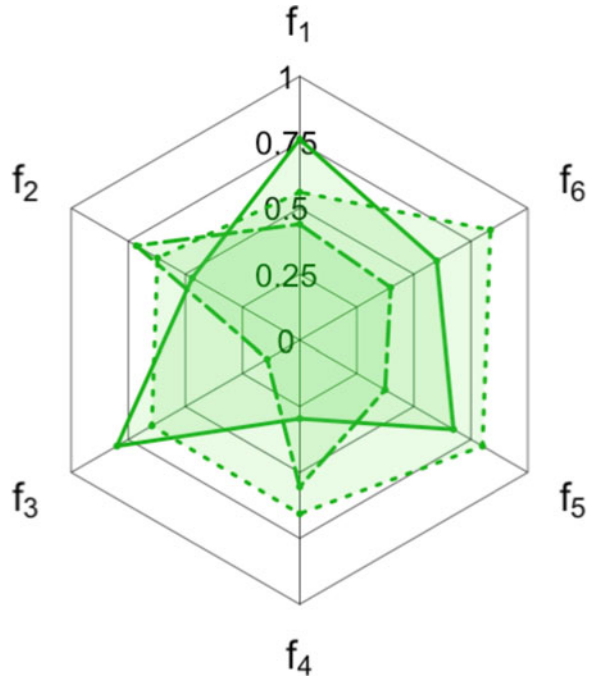
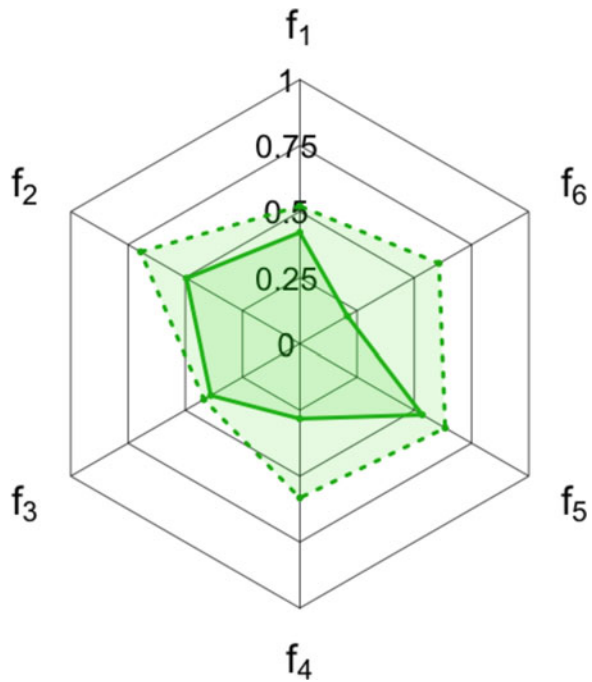


Fig. 8.23 The domination relation in glyph plot. The solid glyph dominates the dashed glyph



more important because instead of a single solution, we are looking for a set of them—the Pareto front—which increases the number of evaluations needed.

Surrogate modelling is an attempt to describe or approximate an unknown function based on a number of previous samples. In optimization, it is a popular method to approximate an expensive evaluation, replacing it with a much cheaper function. A more detailed explanation of surrogate modelling can be found in Chap. 12.

In the same vein as the problem described above, typically, EAs require a lot of function evaluations to be successful. The number of function evaluations depends on the problem and on which EA is being used, but normally, the number of evaluations ranges from hundreds to thousands. Often, the available budget is not enough to evaluate the objective that many times, and surrogate models have to be used. This leads to methods known as surrogate-assisted EAs.

Surrogate-assisted EA for a single-objective problem is quite straightforward; it replaces the objective function with a surrogate model. However, when we move on to multi- and many-objective problems, there is a complication: how do we make the surrogate model for several objectives? This section provides an answer to that question, describing surrogate-assisted EAs applied in multi- and many-objective problems.

In this section we will use the abbreviation MOP for multi- and many-objective problems because surrogate model can be applied to both classes of problems in the same way.

8.5.1 *ParEGO*

ParEGO [32] is a modification from a single-objective surrogate-assisted method named EGO [30] (efficient global optimization). Originally, EGO was proposed to handle single-objective problems with expensive evaluation. ParEGO stands for Pareto EGO, which clearly indicates how it is different from the original EGO: it searches for the Pareto front instead of a single optimum point.

To explain ParEGO, it is better if we start by describing what EGO does. EGO is intended to be used on expensive black-box functions. EGO starts with a set of initial designs obtained from Latin hypercube sampling, one of the several available sampling methods. Based on these initial samples, the EGO creates a DACE kriging model [30, 32], i.e., we try to fit the unknown function to a standard model. Using the model, a new design is suggested based on its **expected improvement**.

Expected improvement is the expectation of a random variable called **improvement** I by Jones et al. [30]. From the initial samples that we have, we will obtain our initial best design which has smallest or largest objective value (for minimization and maximization problem, respectively). Let us call this best design \mathbf{x}^* and its objective value $f(\mathbf{x}^*)$. The DACE kriging model will then provide a prediction and standard error on all possible design points \mathbf{x} . We then treat the objective function $f(\mathbf{x})$ as a normally distributed random variable Y with mean and standard deviation

provided by the DACE predictor and standard error, respectively. The improvement $I(\mathbf{x})$ is calculated from Y using the following equation:

$$\begin{aligned} I(\mathbf{x}) &= \max(f(\mathbf{x}^*) - Y, 0) && \text{for minimization problem} \\ I(\mathbf{x}) &= \max(Y - f(\mathbf{x}^*), 0) && \text{for maximization problem} \end{aligned} \quad (8.7)$$

The expected improvement is then obtained simply by taking the expectation of $I(\mathbf{x})$: $E(I(\mathbf{x}))$.

EGO uses the point with the largest expected improvement, evaluates the objective value, and then rebuilds the kriging model with the new design point. We can opt to use all evaluated designs or take a smaller set, as long as we build a new kriging model every time a new point is evaluated. The process is repeated in a loop until a stopping criteria is met (it can be minimum expected improvement, maximum iteration, etc.).

Now, ParEGO extends this to MOP by using an aggregation method. A set of weight vectors W is built at the start; at each iteration, one of the weight vectors is picked randomly and used for aggregation. At each iteration, different weight vectors will be used but will always be from the set W . The original ParEGO uses the augmented Tchebycheff aggregation method (see [32] for details). As the problem is now transformed into single-objective problem, we can follow the original EGO algorithm for further steps.

To summarize, the ParEGO implementation is nearly the same as the original EGO, except that at every iteration, the objective function is changed (due to the changes in aggregation weights). The changing objective function is not a problem because EGO works on a black-box function, i.e., it does not care what the objective function is, it only cares for the objective values.

8.5.2 Prescreening Method

Another approach of using surrogate model in (evolutionary) MOP is by applying the surrogate and exact evaluation tool in cooperation [20]. In ParEGO, the optimization searches the design space for the best improvement solely based on the surrogate model; the exact evaluation is only used for enriching the data. ParEGO-like methods use the surrogate model as the optimizer.

In the prescreening method, the exact evaluation is used to improve efficiency for local search. In essence, it uses a traditional optimizer, guided by the surrogate model. The role of the surrogate model as a guide is achieved similarly with what happened in ParEGO-like methods which search for points with “improvement.” In ParEGO, the expected improvement is used, but other improvement metrics are actually available as mentioned in [20]. These improvement metrics are then used to **prescreen** the offspring, i.e., pick several candidate-offspring with maximum improvement metrics. This effectively reduces the number of required exact eval-

uations. After the offspring are preselected, the optimizer continues as it would normally. The algorithm is presented in Algorithm 6 [20].

Algorithm 6 Prescreening algorithm

```

 $t = 0$ 
 $P(t) \leftarrow$  Initial population with size  $\mu$ 
Evaluate  $P(t)$  and database of evaluated points  $D$ 
while Stopping criteria not fulfilled do
  while  $|P'(t)| < \lambda$  do
     $P'(t) \leftarrow$  variation  $P(t)$ 
    Evaluate  $P'(t)$  using surrogate model based on  $D$ 
    while  $|P''(t)| < \lambda'$  do
       $P''(t) \leftarrow P'_{best}(t), P'_{best}(t)$  : member of  $P'(t)$  with max improvement
     $P(t+1) \leftarrow$  selection from  $Q \cup P''(t)$ 
   $t = t + 1$ 

```

8.5.3 Taxonomy of Surrogate Models for MOP

As we have seen above, introducing surrogate modelling to MOP can be done in many ways. Deb et al. [14] classified the methods by how many surrogates are used to treat the objectives and constraints. Deb classified them into six groups with the first two groups having two sub-groups (Tables 8.1 and 8.2).

Independent means that for each objective/constraint, one surrogate model is built. In an M -objective problem with N constraints, $M + N$ surrogate models are built for class M1-1. Combined means that only one surrogate model is built for the objectives/constraints, e.g., for class M1-2 $M + 1$ surrogates are built, M for the objectives and 1 for the constraints. This can be done by aggregation or other scalarizing methods.

The optimization method differentiates how the optimization loop looks for the best solution. A decomposed method makes an aggregation for the objectives and

Table 8.1 Taxonomy of the surrogate models in many-objective optimization

Obj. treatment	Cons. treatment	Opt. method	Class
Independent	Independent	Decomposed	M1-1
Independent	Independent	Multi-objective	M1-2
Independent	Combined	Decomposed	M2-1
Independent	Combined	Multi-objective	M2-2
Combined	Independent	1 combined-objective	M3
Combined	Combined	1 combined-objective	M4
Together		Decomposed	M5
Together		Multi-objective	M6

Table 8.2 Available shape and transformation function in the WFG test kit

<i>Shape</i>	<i>Bias</i>
1. Linear	1. Polynomial
2. Concave	2. Flat region
3. Convex	3. Parameter dependent
4. Mixed concave/convex	
5. Disconnected	
<i>Shift</i>	<i>Reduction</i>
1. Linear	1. Non-separable
2. Deceptive	2. Weighted sum
3. Multi-modal	

transforms the problem into several single-objective optimization problems. In turn, the optimization loop needs to be run several times to find different points on the Pareto front. A multi-objective method tries to find the Pareto front simultaneously. Most evolutionary algorithms can be used in this class.

For class M3 and M4, because the objectives are combined into a single surrogate model, we cannot treat it as a multi-objective problem. In these classes, the surrogates are built after the objectives are scalarized.

For class M5 and M6, only one surrogate model is built. In M5, one surrogate model is used to find one point in the Pareto front, while in M6 the single surrogate model is used to find multiple Pareto optimum solutions.

Regarding the taxonomy, ParEGO would fall into class M1-1, while prescreening would fall to M1-2.

8.6 Test Problems for Many-Objective Optimization

As the field is becoming more and more researched, many new algorithms are proposed. To assess the quality of these algorithms, some benchmarking methods are needed. This is done by means of academic test functions.

The test functions are designed as functions that have their Pareto front and Pareto set known in advance or easy to generate, but believed to give optimization algorithms some degrees of difficulties. The difficulties can stem from noise, deceptiveness, bias, etc. [9, 27].

8.6.1 Biobjective Test Problems

We start with problems commonly used for testing multi-objective optimization algorithms, specifically biobjective problems. The ZDT and Black-Box Optimization Benchmarking (BBOB) test problems are well known and fall to this category.

8.6.1.1 ZDT

ZDT is proposed by Zitzler, Deb, and Thiele [47] hence its name. It consists of six test functions with same structures but different shapes and difficulties. The problems are defined as follows:

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} && (f_1(x_1), f_2(\mathbf{x})) \\
 & \text{subject to} && f_2(\mathbf{x}) = g(x_2, \dots, x_m)h(f_1(x_1), g(x_2, \dots, x_m)) \\
 & \text{where} && \mathbf{x} = (x_1, x_2, \dots, x_m)
 \end{aligned} \tag{8.8}$$

The structure above says that the first objective is dependent only on the first parameter, while the second objective can be affected by all parameters (due to the h function).

Each of the six ZDT problems has a different number of parameters m and also different sets of underlying function f , g , and h . However, the problems are designed such that the true Pareto front can always be found when $g(\mathbf{x}) = 1$. With this information, the true Pareto fronts can be generated and be used to assess performance of optimization methods.

8.6.1.2 Black-Box Optimization Benchmarking

Black-Box Optimization Benchmarking (BBOB) is a collection of commonly used test problems. The test problems are categorized in the BBOB function definition [25] to indicate their difficulty factors. However, as the name suggests, even though the functions and their derivatives are known, the test problems should be treated as black-box functions. Black-box functions are mappings in which the users do not know their inner working. The users only interaction with the functions is giving a set of input variables and receiving the output data.

The original BBOB consists of single-objective problems. Two objective problems are also defined in the biobjective BBOB by combining certain pairs of the single-objective problems. The Pareto front of all the biobjective test problems are known and easy to construct or take samples.

The interesting feature of BBOB is any user can use the test problems, report their results, and compare it with the current best algorithm of the specific problem. In other words, the BBOB allows the user to do benchmarking against the best and possibly becoming the new best method.

8.6.2 Scalable Test Problems

With the rise of research on many-objective optimization methods, the biobjective test problems become obsolete because although the number of parameters can be tuned, the number of objectives is fixed in the problems.

Deb et al. [13] proposed that test problems should have scalability in both objective and design spaces and be easy to implement and the real Pareto front must be easy to comprehend. Scalability means that the test problems should have both their number of objectives and number of variables easily tunable. If the test problems are scalable, algorithms can be tested on standard/test cases with varying dimensionality easily.

8.6.2.1 DTLZ

As a trivia, DTLZ is developed by the same group who suggested the ZDT test functions. It is also an acronym of their name with the addition of Laumanns.

The DTLZ [13] test suite consists of seven test problems. Each of the test problems is designed to be able to take the number of objectives and number of variables as an input to construct the complete problem, making it scalable. Each of the seven test problems has different characteristics; for example, DTLZ1 has a linear Pareto front, while DTLZ2 to DTLZ4 have convex forms. Other problems introduce different complicating features such as degenerated or disconnected Pareto front.

The interesting feature of all DTLZ test problems, aside from its scalability, is that the real Pareto fronts are easy to construct. Examples of the DTLZ real Pareto fronts are shown in Fig. 8.24 (DTLZ1) and Fig. 8.25 (DTLZ2, DTLZ3, DTLZ4).

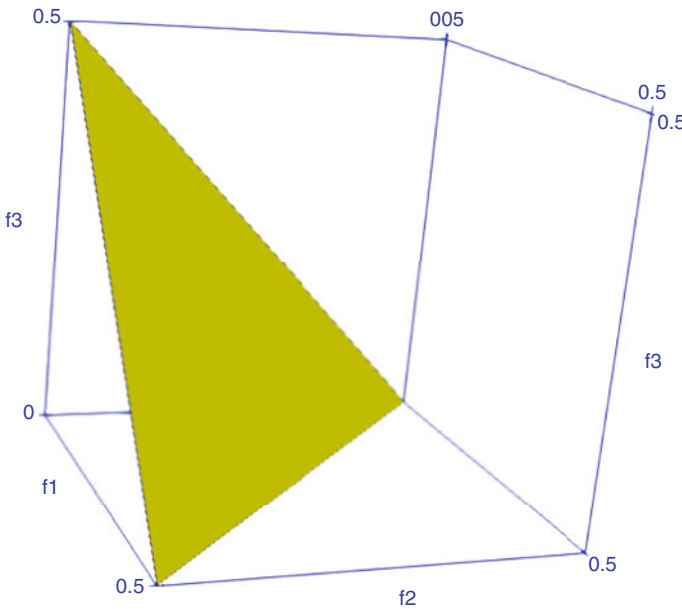


Fig. 8.24 Pareto front for DTLZ1 test problem in 3D objective space. The Pareto front always lies on the hyperplane: $\sum f(x) = 0.5$, and all objective values are positive

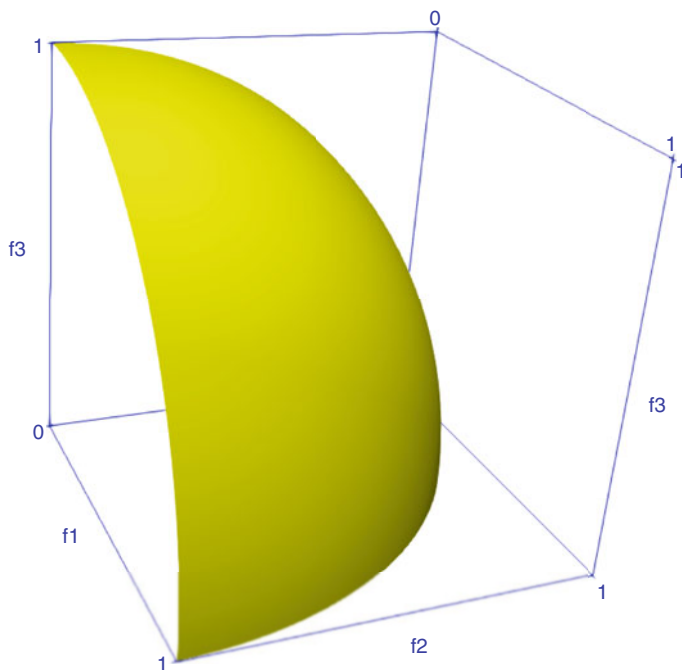


Fig. 8.25 Pareto front for DTLZ2 to DTLZ4 test problems in 3D objective space. The Pareto front lies on the hyperplane $\sum f^2(x) = 1$ (hypersphere with radius 1) and all objective values at positive. This is also the real Pareto front for WFG4 to WFG9 (after normalization)

Because the Pareto fronts are easy to construct, it is also easy to obtain reference points on the Pareto front to calculate performance metrics that requires references on the Pareto front, such as the GD in Sect. 8.3.2.2.

8.6.2.2 WFG

The WFG test kit [26] is a relatively recent work on test problems done by the Walking Fish Group. The WFG test kit has an interesting feature: on top of scalability, the WFG suggests that the difficulties of test problems should also be tunable. The WFG test kit allows users to tune the test problems by adding transformation functions which should increase their difficulties.

The WFG test kit is made of some building blocks called the **shape functions** and **transformation functions**. The shape function is the test functions' base shape. Each of the objectives can have one of the shape functions, i.e., all objectives can have different shapes, but only use one shape function for each objective f_i . The transformation function will then transform (i.e., make changes to) the base shape in form of bias, shift, and reduction.

Although completely tunable, it would not make sense if all researchers use different problems, all tuned by themselves, to compare algorithms' performances. To address this, the WFG also created nine standard test problems (collectively termed as "WFG test suite") so researchers can use this for comparing algorithms.

The nine standard test problems have predefined shape and transformation functions. Users can still tune the number of objectives and variables to some extent (there are still some requirement on the variables).

8.7 Summary

In this chapter we learned about the difference between single-, multi-, and many-objective optimization problems. We also mentioned the challenges which makes many-objective problems much more difficult to solve than multi-objective problems. We also learned the basics of evolutionary algorithms and the building blocks: recombination, mutation, and selection.

Furthermore, we learned how evolutionary algorithms can solve multi- and many-objective optimization problems. Several well-known algorithms were explained. For multi-objective problems, NSGA-II and SMS-EMOA can be used. Both algorithms use non-dominated sorting as their primary selection operator. Their secondary selection operator differs. NSGA-II uses crowding distance, while SMS-EMOA uses S-metric selection. Another difference is that NSGA-II uses $(\mu + \mu)$ selection scheme, while SMS-EMOA uses the steady-state $(\mu + 1)$ scheme.

For many-objective problems, MOEA/D and NSGA-III can be used. The first algorithm, MOEA/D, decomposes the many-objective problem into many single-objective problems. The second algorithm, NSGA-III, is an algorithm based on NSGA-II. The difference lies in the secondary selection method: instead of crowding distance, NSGA-III uses reference points which represent the preferences and weights of different objective functions. Surrogate models which are used in multi- and many-objective problems were also mentioned in this chapter.

In the last section, benchmarking problems for comparing algorithms are explained. These benchmarking problems are predefined functions which are intended to test how well an algorithm can solve problems with different characteristics and difficulties. Some test problems are designed to be scalable, meaning that the number of objective functions and variables can be changed. This feature is especially valuable for researches in many-objective optimization.

References

1. R.B. Agrawal, K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space. *Complex Systems* **9**,115–148 (1994)
2. N. Andreasson, A. Evgrafov, M. Patriksson, E. Gustavsson, M. Onnheim, *Introduction to Continuous Optimization*, 2nd edn. (Studentlitteratur AB, Lund, 2013)

3. J. Arabas, Z. Michalewicz, J. Mulawka, Gavaps-a genetic algorithm with varying population size, in *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, June 1994, vol. 1, pp. 73–78
4. T. Bäck, D.B. Fogel, Z. Michalewicz (eds.) *Handbook of Evolutionary Computation*, 1st edn. (IOP Publishing Ltd., Bristol, 1997)
5. T. Bäck, U. Hammel, H.-P. Schwefel, Evolutionary computation: comments on the history and current state. *IEEE Trans. Evol. Comput.* **1**(1), 3–17 (1997)
6. T.F. Banchoff, D.P. Cervone, An interactive gallery on the internet: “surfaces beyond the third dimension”. *Int. J. Shape Model.* **05**(01), 7–22 (1999)
7. S. Chand, M. Wagner, Evolutionary many-objective optimization: a quick-start guide. *Surv. Oper. Res. Manage. Sci.* **20**(2), 35–42 (2015)
8. C.A. Coello Coello, M. Reyes Sierra, A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in *MICAI 2004: Advances in Artificial Intelligence*, ed. by R. Monroy, G. Arroyo-Figueroa, L.E. Sucar, H. Sossa (Springer, Berlin/Heidelberg, 2004), pp. 688–697
9. K. Deb, Multi-objective genetic algorithms: problem difficulties and construction of test problems. *Evol. Comput.* **7**(3), 205–230 (1999)
10. K. Deb, M. Goyal, A combined genetic adaptive search (genes) for engineering design. *Comput. Sci. Inf.* **26**, 30–45 (1996)
11. K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**(4), 577–601 (2014)
12. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
13. K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in *Congress on Evolutionary Computation (CEC 2002)* (IEEE Press, New York, 2002), pp. 825–830
14. K. Deb, R. Hussein, P. Roy, G. Toscano, Classifying metamodeling methods for evolutionary multi-objective optimization: first results, in *9th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2017*, New York, NY, vol. 10173 (Springer-Verlag New York, Inc., New York, 2017), pp. 160–175
15. K.A. DeJong, *Evolutionary Computation: A Unified Approach* (MIT Press, Cambridge, MA, 2006)
16. M. Ehrgott (ed.) *Multicriteria Optimization* (Springer, Berlin, 2000)
17. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, 1st edn. Natural Computing Series (Springer, Berlin, 2003)
18. M. Emmerich, A. Deutz, *Multicriteria optimization and decision making principles, algorithms and case studies* (2006)
19. M. Emmerich, N. Beume, B. Naujoks, An emo algorithm using the hypervolume measure as selection criterion, in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005 Proceedings*, Guanajuato, March 9–11, 2005, ed. by C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Springer, Berlin, 2005), pp. 62–76
20. M. Emmerich, K.C. Giannakoglou, B. Naujoks, Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Trans. Evol. Comput.* **10**(4), 421–439 (2006)
21. M. Farina, P. Amato, On the optimal solution definition for many-criteria optimization problems, in *2002 Annual Meeting of the North American Fuzzy Information Processing Society Proceedings. NAFIPS-FLINT 2002 (Cat. No. 02TH8622)* (2002), pp. 233–238
22. D.B. Fogel, Nils barricelli - artificial life, coevolution, self-adaptation. *IEEE Comput. Intell. Mag.* **1**(1), 41–45 (2006)
23. T. Glasmachers, B. Naujoks, G. Rudolph, Start small, grow big? Saving multi-objective function evaluations, in *Parallel Problem Solving from Nature – PPSN XIII*, ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (Springer International Publishing, Cham, 2014), pp. 579–588

24. M.P. Hansen, A. Jaszekiewicz, Evaluating the quality of approximations to the non-dominated set (1998)
25. N. Hansen, S. Finck, R. Ros, A. Auger, Real-parameter Black-Box optimization benchmarking 2009: noiseless functions definitions. Research Report RR-6829, INRIA, 2009
26. S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in *Evolutionary Multi-Criterion Optimization*, ed. by C.A. Coello Coello, A. Hernández Aguirre, E. Zitzler (Springer, Berlin, 2005), pp. 280–295
27. S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **10**(5), 477–506 (2006)
28. D. Irawan, Geometrical optimization of plate-fin heat-sink. Master's thesis, Chalmers University of Technology, Gothenburg, 2016
29. H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: a short review, in *IEEE Congress on Evolutionary Computation* (IEEE, New York, 2008), pp. 2419–2426
30. D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13**(4), 455–492 (1998)
31. M. Kitti, History of optimization. <http://www.mitrikitti.fi/opthist.html>. Last accessed 05 Apr 2018
32. J. Knowles, ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **10**(1), 50–66 (2006)
33. A. Messac, C.A. Mattson, Normal constraint method with guarantee of even representation of complete pareto frontier. *AIAA J.* **42**(10), 2101–2111 (2004)
34. K. Miettinen, F. Ruiz, A.P. Wierzbicki, *Introduction to Multiobjective Optimization: Interactive Approaches* (Springer, Berlin, Heidelberg, 2008), pp. 27–57
35. T. Okabe, Y. Jin, B. Sendhoff, A critical survey of performance indices for multi-objective optimisation, in *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, vol. 2 (2003), pp. 878–885
36. N. Riquelme, C.V. Lübben, B. Baran, Performance metrics in multi-objective optimization, in *2015 Latin American Computing Conference (CLEI)* (2015), pp. 1–11
37. H. Seada, K. Deb, U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results, in *Evolutionary Multi-Criterion Optimization*, ed. by A. Gaspar-Cunha, C. Henggeler Antunes, C.C. Coello, (Springer International Publishing, Cham, 2015), pp. 34–49
38. T. Tušar, B. Filipič, Visualization of pareto front approximations in evolutionary multiobjective optimization: a critical review and the prosection method. *IEEE Trans. Evol. Comput.* **19**(2), 225–245 (2015)
39. L. van der Maaten, E.O. Postma, H.J. van den Herik, Dimensionality reduction: a comparative review (2008)
40. D.A.V. Veldhuizen, G.B. Lamont, On measuring multiobjective evolutionary algorithm performance, in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, July, vol. 1, (2000), pp. 204–211
41. D.A.V. Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evol. Comput.* **8**(2), 125–147 (2000)
42. P.A. Vikhar, Evolutionary algorithms: a critical review and its future prospects, in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, December 2016, pp. 261–265
43. T. Wagner, N. Beume, B. Naujoks, *Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization* (Springer, Berlin, Heidelberg, 2007), pp. 742–756
44. M. Wahde and Dawsonera (e-book collection), *Biologically Inspired Optimization Methods: An Introduction* (WIT Press, Southampton, 2008)
45. W.M. Wang, X.Q. Yan, C.W. Fu, A.J. Hanson, P.A. Heng, Interactive exploration of 4d geometry with volumetric halos (2013)
46. Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)

47. E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.* **8**(2), 173–195 (2000)
48. E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2), 117–132 (2003)

Chapter 9

Multilevel Optimisation



Margarita Antoniou and Peter Korošec

Abstract This chapter is a short introduction to multilevel optimisation problems. The simplest multilevel problem is the one that has two levels, where one optimisation problem has as part of its constraints a second optimisation problem, known as bilevel problem. Even this simple version of the problem is from a mathematical point of view, complicated and difficult to solve. Therefore, approaches and examples of bilevel problems, as well as special cases and extensions of this problem that are used widely in literature, are presented. The most common methodologies used to solve multilevel problems are then described, with more extended reference to metaheuristic methods.

Keywords Multilevel optimisation · Bilevel · Hierarchical optimisation · Minimax problem · Metaheuristic methods

9.1 Introduction

The standard optimisation problem is the one that has a single-objective function that needs to be optimised while satisfying some constraints and can be considered as a single-level optimisation problem. Unfortunately, some real-world applications cannot be represented in this way; therefore it has been extended in its form and complexity, so it can have more objectives, different constraints, different types of variable vectors, etc. [1]. One extension is to have multiple levels of optimisation tasks instead of just one. A large number of application problems require more than one level of optimisation, where one optimisation task is nested inside the other [2]. In its simplest form, the nested optimisation problem constitutes two levels. These problems are known as bilevel optimisation problems.

M. Antoniou (✉) · P. Korošec
Jožef Stefan Institute, Ljubljana, Slovenia

Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
e-mail: margarita.antoniou@ijs.si; peter.korosec@ijs.si

The first formulation of bilevel problem was documented in 1973 by Bracken and McGill [3], and the definition of bilevel and multilevel programming was used for the first time some years later by Candler and Norton in [4]. Since then, both classical and evolutionary optimisation communities have studied bilevel optimisation problems. These problems are intrinsically difficult to solve, so it is no surprise that most of the proposed solution methods are either computationally very expensive or applicable only to the simplest cases of bilevel optimisation problems that have some nice mathematical properties [5].

One can find in literature the usage of multilevel optimisation referring to an optimisation approach. While multilevel optimisation problems present a hierarchy in the way the decision-solutions are taken, the multilevel paradigm is used to describe the approach in optimisation that presents a hierarchy in the procedure of the optimisation. The most common practice of solving an optimisation problem is computing/evaluating solutions in some iterative process. In many applications, such as in engineering, FEM or CFD computations, this can become really expensive in terms of computational time. With the multilevel approach, one targets into the minimisation of the expensive evaluations, by allowing for less accurate computing by using computationally cheaper models of the problem or by reducing solutions search space that needs to be explored. Both approaches heavily depend on managing the balance between accuracy and computational time, so the best results are achieved in a shorter time. The general multilevel strategy can be found in the literature with several names and definitions. Though there are some differences in a couple of aspects of each word, the main idea of the hierarchical method is the same. Therefore, we can say that multilevel can be also found as *multi-fidelity*, *multi-scale*, *multi-grid* or *multistage* optimisation approaches. Much more on this topic a reader can find in [6–12].

In the following sections, we will focus only on multilevel and bilevel problems by giving definitions, presenting a simple example of a linear bilevel optimisation and its comparison to biobjective optimisation. Also, special cases of bilevel problems are shown, providing an idea to the reader on how multilevel problems can be found and/or formulated in different applications. Moreover, the main solution algorithms with a special reference to metaheuristic solution methods found in the literature are presented. The chapter ends by giving some examples of applications solved as bilevel optimisation problems.

9.2 Multilevel Optimisation Problem

In the following of the chapter, we will consider minimisation problems without the loss of generality since $\min = -\max$. The general multilevel optimisation problem (P) can be formulated as follows:

$$(P_1) \min_{x_1 \in X_1} f_1(x_1, x_2, \dots, x_k)$$

subject to

$$g_1(x_1, x_2, \dots, x_k) \leq 0$$

where P_2 solves

$$(P_2) \min_{x_2 \in X_2} f_2(x_1, x_2, \dots, x_k)$$

subject to

$$g_2(x_1, x_2, \dots, x_k) \leq 0$$

...

...

where P_k solves

$$(P_k) \min_{x_k \in X_k} f_k(x_1, x_2, \dots, x_k)$$

subject to

$$g_k(x_1, x_2, \dots, x_k) \leq 0$$

where P_1 is called the first (upper) level problem and corresponds to the highest level in the hierarchy; x_1 is the solution, composed of n_1 variables, of the first level problem from the set of solutions $X_1 \in R^{n_1}$; x_2 is the solution, composed of n_2 variables, of the second level problem from the set of solutions $X_2 \in R^{n_2}$; and x_k is the solution, composed of n_k variables, of the k -th level problem from set of solutions $X_k \in R^{n_k}$. At this level, the decision maker controls the decision variables x_1 , and his/her objective is to minimise the function f_1 . Consequently, P_k is the k -th level problem, which corresponds to the lowest level in the hierarchy [13]. Let us assume that each of the levels corresponds to a decision maker, which we call player from now on. Each player has control over its own set of variables and aims to optimise its own objective function f . Each objective function can also depend on the variables of other players [14]. Variable value choices of each player should be such that there exists a sequence of choices for the other, following players that all satisfy their constraints. Players are playing in a hierarchical order, meaning the first player (first level) chooses first and the k -th (last) player plays last.

Multilevel optimisation problem presents a nested formulation. Finding a polynomial algorithm, capable of obtaining the global optimum for even the simplest case of linear optimisation problems with only two levels, is highly unlikely. For this reason, we will refer to instances and solution algorithms of bilevel optimisation for

the rest of the chapter, as we believe that this is the best way for an introduction to multilevel optimisation problems.

9.3 Bilevel Optimisation Problem

From a mathematical point of view, bilevel optimisation problem (BOP) consists of two levels of optimisation tasks. Two different sets of variables belong to each of these tasks. The level corresponds to the hierarchy of the problem, meaning that there exists an upper and lower level optimisation problem. The mathematical representation is as follows:

$$\min_{x_u \in X_u, x_l \in X_l} F(x_u, x_l)$$

subject to

$$G_k(x_u, x_l) \leq 0, \quad k = 1, \dots, K,$$

where K is the number of constraint functions of the upper level and x_l is the solution of the lower level problem from the set of solutions $X_l \in R^n$, with regard to solution from upper level x_u from set of solutions $X_u \in R^m$, according to:

$$\min_{x_l \in X_l} f(x_u, x_l)$$

subject to

$$g_j(x_u, x_l) \leq 0, \quad j = 1, \dots, J,$$

where J is the number of constraint functions of the lower level. F represents the first (upper) level optimisation problem and corresponds to the highest level in the hierarchy. At this level, the decision maker controls the decision variables x_u , and his/her objective is to minimise the function F . Consequently, f represents the second level of the optimisation problem, which corresponds to the lowest level in the hierarchy [13]. Note that in some sections we also use notation $F(x, y)$ for the upper level and $f(x, y)$ for the lower level, where x is the solution of the upper level and corresponds to x_u and y is the solution of the lower level and corresponds to x_l . The basic notations and definitions of a bilevel optimisation problem are the following as found in [5]:

- **Decision vectors:** $x_u \in X_U$ (or $x \in X$) corresponds to the leader's (upper level) decision variable and decision space and $x_l \in X_L$ (or $y \in Y$) corresponds to the follower's (lower level) decision variable and decision space.

- **Objectives:** F is the leader's (upper level) objective functions. f is the follower's (lower level) objective functions.
- **Constraints:** $G_k, k = 1, \dots, K$ are the leader's (upper level) constraint functions. $g_j, j = 1, \dots, J$ are the follower's (lower level) constraint functions.
- **Lower level feasible region:** $\Omega : X_U \implies X_L, \Omega(x_u) = \{x_l : g_j(x_u, x_l) \leq 0 \forall j\}$ represents the lower level feasible region for any given upper level decision vector.
- **Constraint region (relaxed feasible set):** $\Phi = \{(x_u, x_l) : G_k(x_u, x_l) \leq 0 \forall k, g_j(x_u, x_l) \leq 0 \forall j\}$ represents the region satisfying both upper and lower level constraints.
- **Lower level/rational reaction set:** $\Psi : X_U \implies X_L,$

$$\Psi(x_u) = \{x_l : x_l \in \arg \min_{x_l \in X_L} f(x_u, x_l) : x_l \in \Omega(x_u)\},$$

represents the lower level optimal solution(s) for an upper level decision vector.

- **Inducible region (feasible set):**

$$I = \{(x_u, x_l) : (x_u, x_l) \in G_k(x_u, x_l) \leq 0, x_l \in \Psi(x_u)\}$$

represents the set of upper level decision vectors and corresponding lower level optimal solution(s) belonging to feasible constraint region.

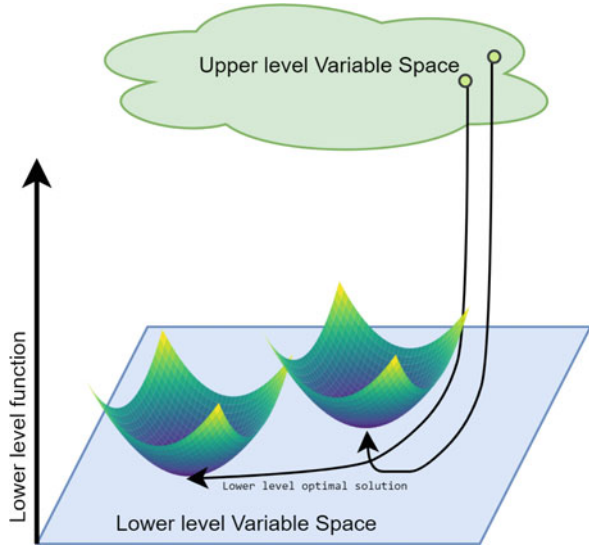
- **Choice function:** $\psi : X_U \rightarrow X_L, \psi(x_u)$ represents the solution chosen by the follower for any upper level decision vector. It becomes important in case of multiple lower level optimal solutions.
- **Optimal solution:** A solution $(x_u^*, x_l^*) \in I$ is an optimal solution if $\forall (x_u, x_l) \in I, F(x_u^*, x_l^*) \leq F(x_u, x_l)$.

A general sketch of the bilevel problem, inspired by [2], can be seen in Fig. 9.1, in which the variable spaces of upper and lower levels are illustrated. For one decision variable of the upper level, one lower level optimal solution is represented.

Bilevel optimisation problems can be also interpreted as non-cooperative static Stackelberg game, as was first introduced by von Stackelberg in 1934 in the context of unbalanced economic markets [15]. A bilevel problem is considered a game where two decision makers follow a hierarchy. The upper and the lower levels are termed as the leader and the follower, respectively. The leader is the first to perform an optimisation step (decision) for his/her objective function. The follower reacts having full knowledge of the leader's choice. The follower's decision, though, affects the leader's decision in an implicit manner, since it changes some of the variables used by the leader [16, 17].

The main characteristic of the bilevel optimisation problem is its nested nature. Hansel et al. have proved that bilevel programming is strongly NP-hard [18]. Moreover, bilevel optimisation problems are typically non-convex and disconnected. In general, solving an optimisation problem produces one or more feasible solutions. In the case that at the lower level there are multiple global optimal solutions,

Fig. 9.1 A general sketch of a bilevel problem, inspired by [2]



bilevel optimisation problem has to cope with additional challenges. With regard to the selection of one solution, two positions are distinguished, according to the assumption that the two levels are cooperating or conflicting [19]. These positions are the following:

- **Optimistic position:** The upper level expects the lower one to choose such a solution from the optimal set that leads to the best objective function value at the upper level. This assumes cooperation between the two levels. Due to its tractability compared to the pessimistic approach, most of the literature treats the bilevel optimisation problem as optimistic.
- **Pessimistic position:** In this case, upper level optimisation is ready for the worst case. The upper level assumes that the lower level will choose a solution from the optimal set that leads to the worst objective function value at the upper level [5].

9.3.1 Linear Bilevel Optimisation Example

For a better understanding of the bilevel optimisation problem, an example of a linear bilevel optimisation problem is presented in this section. Let us consider the following continuous linear BOP, where x represents upper level and y lower level solution:

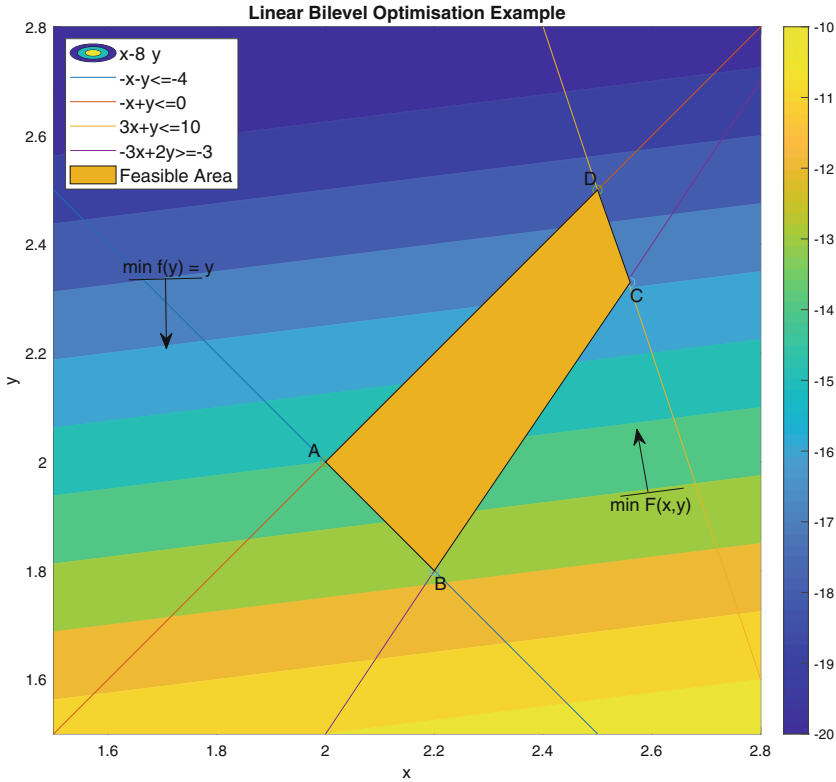


Fig. 9.2 Bilevel optimisation example and its feasible region

$$\left\{ \begin{array}{l} \min_{x \geq 0} F(x, y) = x - 8y \\ \text{subject to} \quad \min_{y \geq 0} f(y) = y \end{array} \right. \quad \text{subject to} \quad \left\{ \begin{array}{l} -x - y \leq -4 \\ -x + y \leq 0 \\ 3x + y \leq 10 \\ -3x + 2y \geq -3 \end{array} \right. \tag{9.1}$$

Figure 9.2 illustrates the feasible region considering the lower level constraints. Additionally, one can see the direction in which the upper level problem is minimising its values. More specifically, the objective function values are shown, as a contour plot with values described in the right contour legend. The inducible region is shown in Fig. 9.3 of the problem which is represented by the lines created from points A-B-C. Along these lines, the optimal solution of the linear BOP is point C (2.56, 2.33), with $F = -16.31$ and $f = 2.33$ as it also minimises the upper

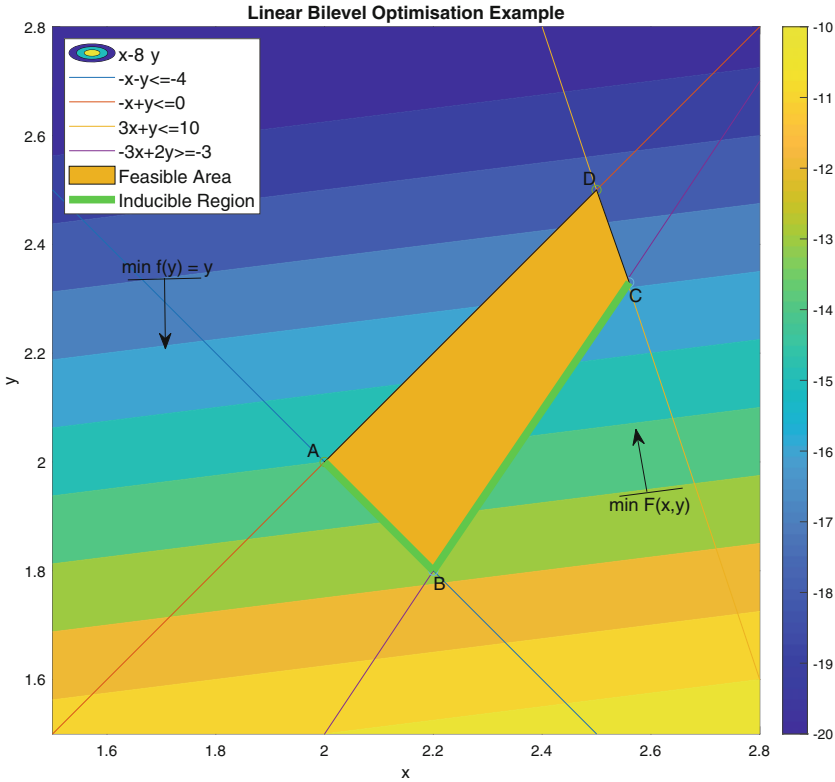


Fig. 9.3 Bilevel optimisation example and its inducible region

level. Point A can be considered as a solution to the bilevel problem, just not the optimal one.

9.4 Bilevel vs Biobjective Optimisation Problem

An optimisation problem can have more than one objective. In this case, we talk about a multiobjective optimisation problem. The one with two objectives is called biobjective optimisation problem and has the following formulation:

$$\begin{cases} \min_{x,y} F(x, y) \\ \text{subject to} & G(x, y) \\ \min_{x,y} f(x, y) \\ \text{subject to} & g(x, y) \end{cases} \quad (9.2)$$

where F and f are the two objectives to be optimised, x and y are the decision variables, and G and g are the constraints of the first and second objectives, respectively. In single-objective optimisation problems, relations between solutions (better/worse) are easily determined through comparison of their objective values. In biobjective optimisation, this is not so obvious. Here the superiority of the solutions is determined by the dominance. According to [1], a solution x_1 is said to dominate x_2 solution, if solution x_1 is no worse than x_2 in all objectives and solution x_1 is strictly better than x_2 in at least one objective. Given this definition, the non-dominated set of solutions in the feasible region¹ is called the Pareto-optimal set, and the boundary marked of the solution objectives values of this set is called Pareto-optimal front.

Many researchers tried to investigate the relationship between the bilevel and biobjective optimisation problem [20, 21]. It has been shown that, while in some special cases and examples the BOP can be formulated as a biobjective optimisation problem and results can be found that way, there are no conditions that an optimal solution of a BOP is in the Pareto-optimal front of its equivalent biobjective optimisation problem.

For better understanding the differences between the two problems a comparison of the two different problems are made, inspired by Talbi in [22], using the example from the previous section. We will show that a BOP does not necessarily have an equivalent corresponding biobjective problem composed of the upper level and the lower level objectives. Consequently, the optimal solution of the BOP is not automatically a Pareto-optimal solution of the biobjective problem and vice versa. For this reason, finding a solution by reformulating the bilevel as biobjective optimisation problem and using the Pareto dominance will not work. Let us reformulate the previous bilevel problem as a biobjective one as follows:

$$\left\{ \begin{array}{l} \min F(x, y) = x - 8y \\ x \geq 0 \\ \min f(y) = y \\ y \geq 0 \\ \text{subject to} \quad -x - y \leq -4 \\ \quad \quad \quad -x + y \leq 0 \\ \quad \quad \quad 3x + y \leq 10 \\ \quad \quad \quad -3x + 2y \geq -3 \end{array} \right. , \tag{9.3}$$

where F and f are the previous upper and lower level objectives, but now they are optimised independently and on a single level.

In Fig. 9.4, a scatterplot of the feasible decision space of the biobjective problem is shown. It is obvious that the feasible space is formulating a trapezium, identical to the one in the bilevel example. Figure 9.5 is a scatterplot of the corresponding

¹Feasible region is consisting of the set of all solutions that satisfy all the constraints.

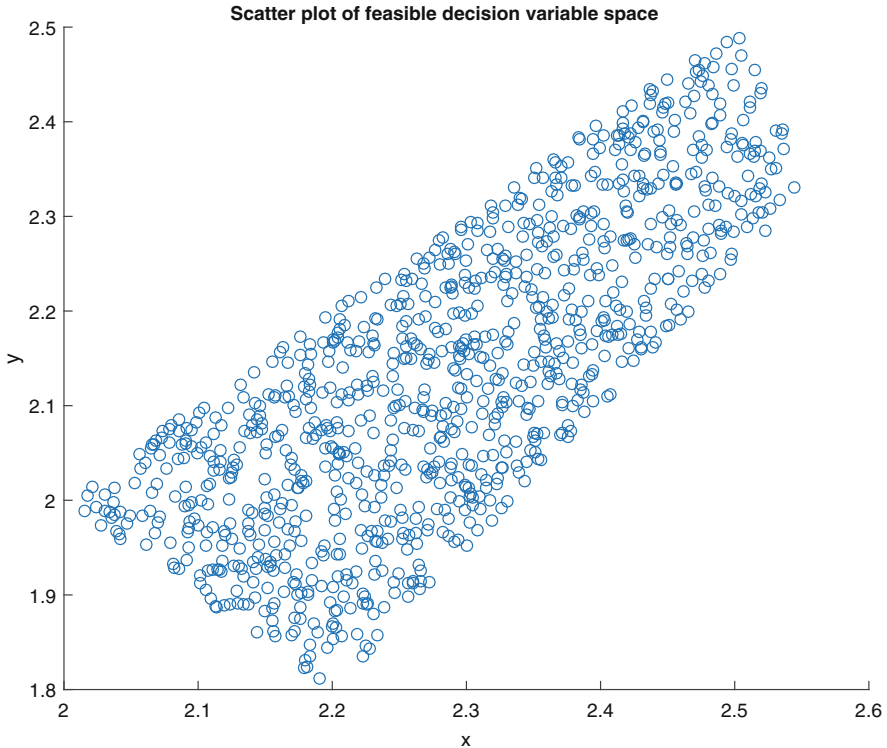


Fig. 9.4 Feasible decision variable space

objective space of the biobjective problem. The red point in Fig. 9.5 inside the feasible objective space corresponds to the optimal solution of the bilevel example. One can easily notice that this solution is dominated by many other solutions to the biobjective optimisation problem. More specifically, the red-filled triangle, shown in Fig. 9.6, which is formed by constraint lines and the upper and lower level objectives, represents the region of solution of the biobjective problem that is dominating the bilevel optimal solution (point C). From this, one can state that bilevel and the corresponding biobjective problem are two completely different concepts and referring to two different kinds of problems.

9.5 Special Cases of Bilevel Optimisation Problems

Bilevel optimisation problems can occur in real-world applications with different formulations and characteristics. A good overview paper that refers to these different cases of BOPs is [17]. In this section the most popular special cases are presented, so the reader can have a general idea of the range of formulations a BOP can have.

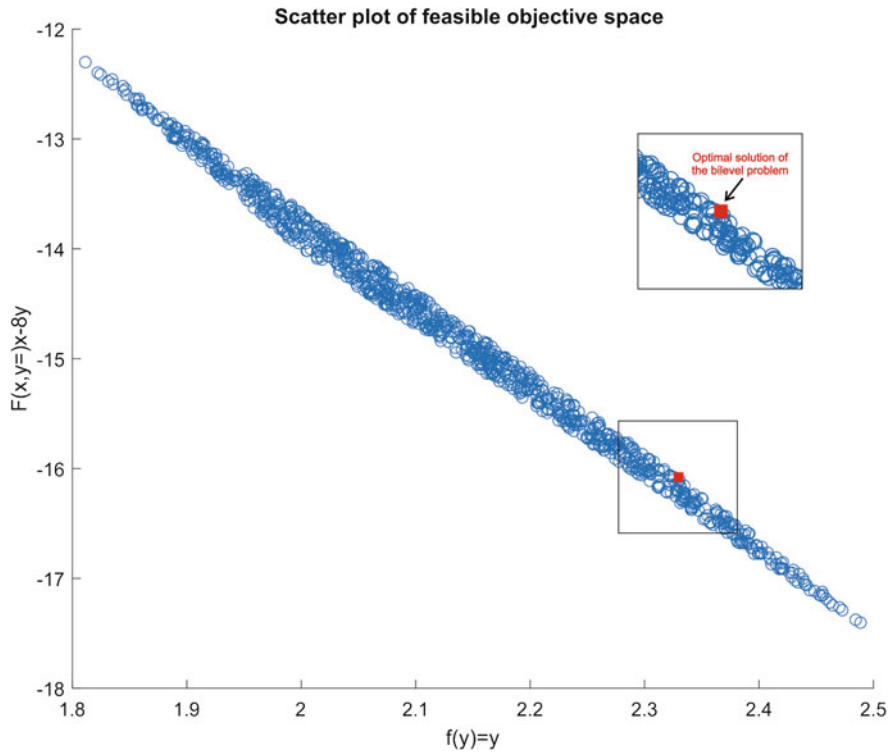


Fig. 9.5 Feasible objective space

Bilevel multiobjective, bilevel multileader/multifollower, bilevel under uncertainty, and minimax as a bilevel optimisation problem are introduced as special cases of the BOP.

9.5.1 Bilevel Multiobjective Optimisation Problems

It is possible that upper and/or lower levels have multiple objectives. The mathematical definition of such problem is:

$$\min_{x_u \in X_U, x_l \in X_L} F(x_u, x_l) = (F_1(x_u, x_l), \dots, F_p(x_u, x_l))$$

subject to

$$x_l \in \arg \min_{x_l \in X_L} (f_1(x_u, x_l), \dots, f_q(x_u, x_l)) :$$

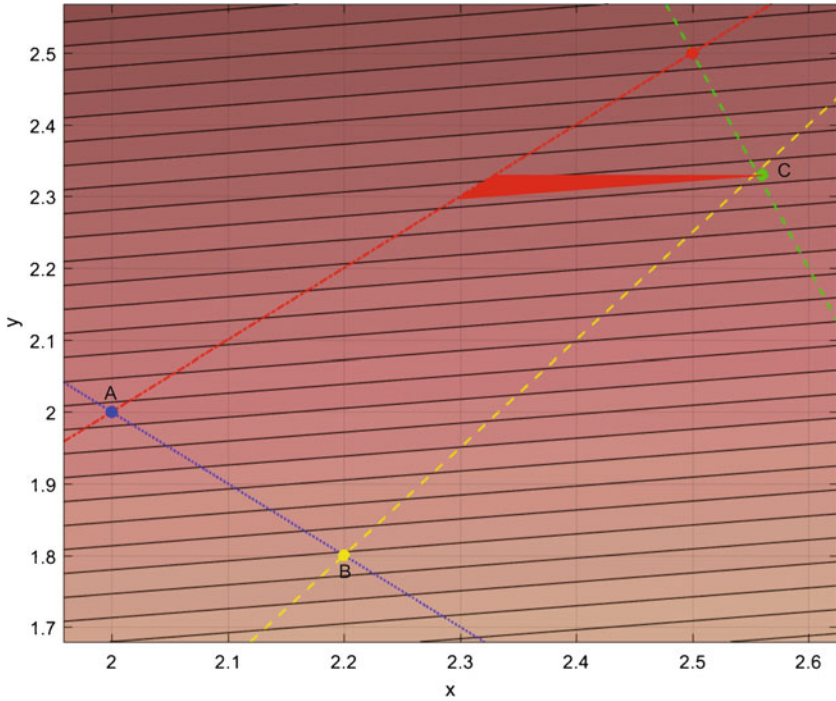


Fig. 9.6 Bilevel vs biobjective optimisation example. The red triangle represents the region of solution of the biobjective problem that is dominating the bilevel optimal solution (point C)

$$g_j(x_u, x_l) \leq 0$$

$$G_k(x_u, x_l) \leq 0$$

where

$$G_k : X_U \times X_L \rightarrow \mathbb{R}, k = 1 \dots, K$$

denote the upper level constraints and K is the number of upper level constraints and

$$g_j : X_U \times X_L \rightarrow \mathbb{R}, j = 1 \dots, J$$

represent the lower level constraints and J is the number of lower level constraints, respectively [5]. F and f are the upper and lower level objectives, respectively, where p is the number of upper level objectives and q is the number of lower level objectives. Also, $X_U \in \mathbb{R}^n$ and $X_L \in \mathbb{R}^m$, where n and m are the number of variables for upper and lower levels, respectively.

For more information on the bilevel multiobjective problems, the reader can refer to [23, 24]. An example of a bilevel multiobjective problem can be found in [25], where it was applied in a transportation system planning problem and solved using genetic algorithm. Another example can be found in [26] where a ‘probabilistic bilevel linear multiobjective programming problem’ and its application in enterprise-wide supply chain planning problem was approached and solved with a fuzzy programming technique.

9.5.2 Bilevel Multileader and/or Multifollower Optimisation Problems

In many applications, a bilevel optimisation problem may involve multiple decision entities (decision makers) on the upper level. The bilevel problem, in this case, has many leaders that may have their individual decision variables, objective functions and/or constraint conditions. This kind of bilevel decision problem is called a bilevel multileader problem [27]. A general definition of the bilevel multileader problem is [17]:

For $x_i \in X_i \subset R^{p_i}$, $y \in Y \subset R^q$, $i = 1, 2, \dots, L$, in which L is the number of leaders and one follower

$$\min_{x_i \in X_i} F_i(x, y)$$

subject to

$$G_i(x, y) \leq 0,$$

where for each $x = (x_1, x_2, \dots, x_L)$ given by the first level, y solves

$$\min_{y \in Y} f(x, y)$$

subject to

$$g(x, y) \leq 0,$$

where L is the number of leaders, while x_i and y are the decision variables of the i -th leader and the follower, respectively. $F_i, f : R^{p_1} \times \dots \times R^{p_L} \times R^q \rightarrow R^1$ are the objective functions of the i -th leader and the follower, respectively. $G_i : R^{p_1} \times \dots \times R^{p_L} \times R^q \rightarrow R^{m_i}, g : R^{p_1} \times \dots \times R^{p_L} \times R^q \rightarrow R^n$ are the constraint conditions of the i -th leader and the follower, respectively. The sets X_i and Y can induce upper and lower bounds on the decision variables, restricting the problem even more. In general, in this problem the leaders have to take into consideration the

reaction of the followers along with the decision results given by their counterparts at the upper level.

A bilevel decision problem may involve multiple decision makers at the lower level, formulating a bilevel multifollower problem. These followers may have different reactions for a possible decision made by the leader. These followers may share decision variables, objectives and/or constraints. The reactions of individual followers and their relationships will affect the leader's decision [27].

A definition of the multifollower problem is [17]: For $x \in X \subset R^p$, $y_i \in Y_i \subset R^{q_i}$, $i = 1, 2, \dots, K$, in which one leader and K followers are involved

$$\min_{x \in X} F(x, y)$$

subject to

$$G(x, y) \leq 0,$$

where for each x given by the upper level, $y = (y_1, y_2, \dots, y_K)$ solves the i -th follower's problem

$$\min_{y_i \in Y_i} f_i(x, y)$$

subject to

$$g_i(x, y) \leq 0,$$

where x and y_i are the decision variables of the leader and the i -th follower, respectively. $F, f_i : R^p \times R^{q_1} \times \dots \times R^{q_K} \rightarrow R^1$ are the objective functions of the leader and the i -th follower, respectively and $G : R^p \times R^{q_1} \times \dots \times R^{q_K} \rightarrow R^m$, $g_i : R^p \times R^{q_1} \times \dots \times R^{q_K} \rightarrow R^{n_i}$ the constraint conditions of the leader and the i -th follower, respectively. In this problem, the followers have to take into consideration the decision results of their counterparts when making their individual decisions, after knowing the leaders' decision.

In literature, these problems have been accounted for problems with several leaders and one follower, one leader and more followers and combination with more than one leaders and followers. Papers tackling these problems are [28–30].

9.5.3 Bilevel Optimisation Problem Under Uncertainty

Decision variables of bilevel optimisation problems can present uncertainties at both levels. Each level has different importance to the overall result of a bilevel optimisation problem. Therefore, the impact of the uncertainties of upper and lower level variables on the final robust solution of the problem is expected to be different,

too. The nested nature of the bilevel optimisation problems makes the search of robust solutions substantially more challenging compared to single-level optimisation problems [5, 31]. Suggested definitions of bilevel optimisation problems under uncertainties and ideas for tackling variable uncertainty are presented in [31]. Test problems can be found in [32].

9.5.4 *Minimax (Worst-Case Scenario) as Bilevel Optimisation Problem*

In many real-world applications, uncertainties occur that are impossible to avoid, and they might be due to several reasons such as reduced accuracy of the simulations, manufacture tolerances, unknown conditions of the problem, etc. A way to manage these uncertainties is robust optimisation², by taking into consideration these uncertain parameters. One way to do this is by transforming them into decision variables. Having that said, these uncertain parameters must be optimised so that they can respond to the worst-case scenario they are able to describe (where as scenario we mean a specific realisation of all parameters of the problem) [34].

In the worst-case scenario, the set of possible scenarios is described, and the objective is to find a solution that performs reasonably well for all scenarios. This solution is the one that has the best “worst-case” performance and performs well even in the most hostile scenario [35].

Using minimax [36] as an approach to tackling uncertainty, the optimisation of the worst-case scenario can be translated into solving a bilevel optimisation problem where the worst case is defined as a maximum in the uncertain space (uncertain parameters) and the optimal design corresponds to the minimum of all the maxima in the design space (candidate solutions) [37, 38].

The minimax problem can be described as:

$$\min_{x \in X} f(x, y)$$

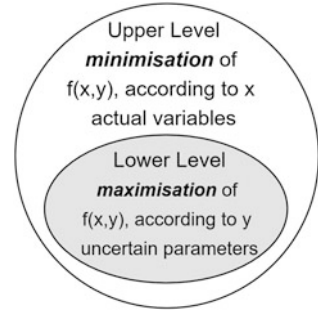
where

$$f(x, y) = \max_{y \in Y} f(x, y)$$

and $X \in R^m$ represent the set of candidate solutions and $Y \in R^n$ the set of all possible scenarios [39]. Note that the upper and the lower levels have the same objective function $f(x, y)$, where the upper level is minimising according to the actual variables x and the lower level maximising according to the uncertain

²Note that this is one approach of robust optimisation. A practical guide for robust optimisation is given by [33].

Fig. 9.7 Minimax optimisation (worst-case scenario)



parameters y of the problem. If the actual problem is a minimisation problem, then the worst-case scenario given by the uncertain variables y can be found by maximising $f(x, y)$. The minimax optimisation schema is shown in Fig. 9.7.

In the literature, one can find many approaches to solving this problem. Lung et al. [39] approached minimax optimisation problem by means of evolutionary algorithms, using a differential evolution algorithm for numerical optimisation. Recently, Ortega et al. [38] proposed a novel heuristic to solve multiobjective minimax problems, approximating the worst-case Pareto-optimal front at a very reduced cost with respect to approaches based on nested optimisation. Zhou et al. [40] proposed a surrogate-assisted evolutionary algorithm for tackling minimax optimisation problems. In this minimax algorithm, a surrogate model based on the Gaussian process is built to approximate the relationship between the decision variables and the objective value. In each generation, most of the new solutions are evaluated using the surrogate model, and only the best one is evaluated by the actual objective function, reducing the computational cost and time. In [34] a variable neighbourhood search algorithm is proposed and was applied to a black box wing-shaped optimisation problem, giving good preliminary solutions. An example of a constrained minimax approach can be found in Chap. 17 of the OTS book.

9.6 Solution Algorithms

Bilevel optimisation problems have been studied widely by the scientific community, both from a theoretical and mathematical approach, and more recently, by the metaheuristic and evolutionary research community. In this section, we will focus on the metaheuristic approaches and algorithms that have been developed to solve bilevel optimisation problems, while some of the classical approaches will be mentioned.

9.6.1 *Classical Approaches*

Well-behaved bilevel optimisation problems are formulated in the literature in order to be solved with classical approaches, due to their inherent complexity. Assumptions about the nature of the equations, such as that they are linear, quadratic or convex, are made. The main classical approaches are listed, along with some references for the reader that wants to find out more about the topic, as a more detailed report of them is out of the scope of this chapter.

- Single-level reduction [16]
- Smoothing methods [16]
- Descent algorithm [41]
- Penalty methods [42]
- A trust region method [43]

9.6.2 *Metaheuristic Approaches*

Many metaheuristics algorithms have been so far applied to bilevel optimisation, with most of the approaches using nested strategies. Evolutionary algorithms (EAs), a popular subset of metaheuristics, are an efficient way of solving bilevel optimisation problems since no implementable mathematical optimality conditions exist [44]. To use classical numerical optimisation methods, various simplifying assumptions are made, such as continuity, differentiability and convexity of the problem. These assumptions though do not represent most of the real-world problems. EA's population-based approach, along with the flexibility of its operators and without the need for simplifying the objective functions, leads to acquiring better solutions of the BOPs [45]. Most of the times, the solutions of the lower/upper level are multimodal, making the use of surrogate modeling very promising, since a simpler approximation of the problem is solved, making the procedure faster [5]. Last but not the least, many bilevel optimisation problems are multiobjective, where both levels require to find and maintain multiple optimal solutions, and EAs are known to be good for these cases [19]. According to the classification of Sinha et al. [5] and Talbi [22], the metaheuristic approaches can be categorised as follows:

- Nested methods: the lower level optimisation problem is solved in a nested way, meaning that for each upper level solution, a lower level solution is obtained and evaluates the solutions generated at the upper level of the BOP. In nested strategies for every upper level solution, a lower level optimisation task is executed, as seen in Fig. 9.8. This usually means that they are computationally expensive and not practical for large-scale bilevel optimisation problems. We refer the reader to the following papers [46, 47] for more detailed information.
- Single-level reduction: the BOP is reduced into a single-level optimisation problem, and then any traditional evolutionary algorithm can be used to solve the

Fig. 9.8 Nested method

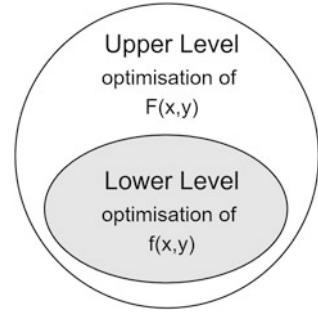
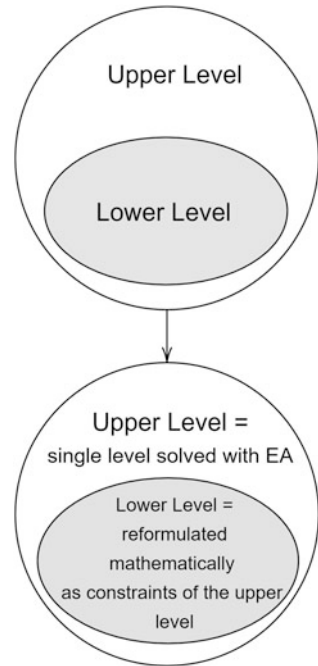


Fig. 9.9 Single level reduction method



problem. A general schema is shown in Fig. 9.9. Karush-Kuhn-Tucker (KKT) conditions³ applied to the lower level have been often used in the evolutionary computation community to reduce the bilevel optimisation problem into a single-level optimisation problem. This approach is able to solve problems that adhere

³Karush-Kuhn-Tucker (KKT) conditions: when the lower level problem has some conditions, such as that is convex and regular, it can be replaced with its KKT conditions. KKT conditions on the lower level problem are generally used as constraints in the formulation of the KKT conditions of the upper level optimisation problem [45]. This involves the second derivatives of the objectives and constraints of the lower level problem as necessary conditions of the upper level optimisation problem [22]. That way, the problems are reduced to a single-level optimisation problem. More about this the reader can find in [48].

to certain regularity conditions at the lower level because of the requirement of the KKT conditions. The upper level objective function and constraints, however, can be more general and complex, as the reduced single-objective problem is solved with an evolutionary algorithm. For example, Wang et al. reduced the bilevel optimisation problem into a single-level optimisation problem using KKT conditions and tested the algorithm in a number of standard test problems [49]. The algorithm managed to handle non-differentiability only at the upper level objective function. Later, Wang et al. improved this algorithm, enabling it to handle non-convexity of the lower level problem and thus obtaining better results than their previous approach [50]. A serious drawback of both methods though is the high number of function evaluations (requiring 100,000 function evaluations for 2–5 variable bilevel optimisation problems).

- **Meta-modeling-based methods:** Meta-modeling-based methods are usually used for optimisation problems when each true evaluation—meaning the exact solution—is computationally expensive. Meta-model or surrogate model can be defined as an approximation function of an actual model, which is simpler and easier to evaluate. The surrogate model is trained and used in the optimisation, based on a limited sample of true evaluations of the actual model. A relevant review on the use of fitness approximation in the context of evolutionary computation has been reported by Yin [47]. Since the bilevel optimisation problems are inherently complex, the number of evaluations needed is large, making meta-modeling, especially when used with population-based algorithms, a very promising alternative [2]. Algorithms published until today, according to Islam et al. [51], that use approximation models during the evolutionary bilevel search are shown in Table 9.1.
- **Co-evolutionary approaches:** these approaches constitute the most general methodology to solve multilevel optimisation problems. Many metaheuristics, usually one for each level, are solving different levels of the problem in parallel while they exchange information between them during the optimisation process.

Table 9.1 Meta-modeling-based algorithms

Meta-modeling-based methods	
Modified NBLEA [52]	NBLEA is a basic nested algorithm in which each of the levels is optimised using an EA and the lower level problem is first modeled as a quadratic programming problem
BLEAQ [53]	BLEAQ is an efficient bilevel EA based on quadratic approximations, where the optimal lower level variable values are approximated as a function of the upper level variables
Surrogate-assisted BIDE [54]	BIDE uses an extensive differential evolution algorithm at both levels, with a surrogate model that is built by approximating the relationship between the upper level variables and the corresponding lower level optimum
SABLA [51]	SABLA uses surrogate models of multiple types, in order to provide the flexibility of approximating different types of functions more accurately

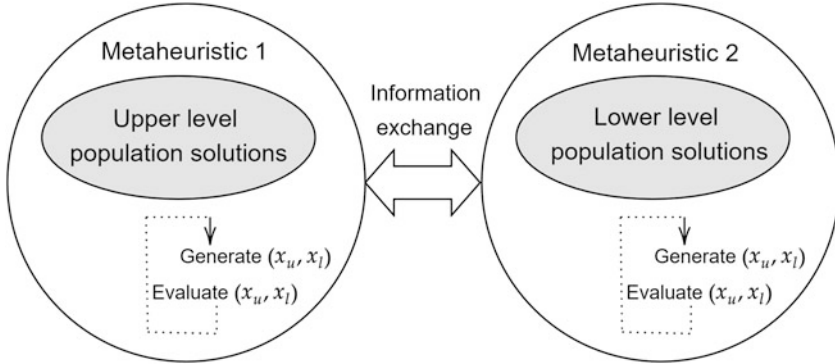


Fig. 9.10 Co-evolutionary approach for solving BOPs. The two metaheuristics evolve in parallel and cooperate via information exchange [22]

Since in BOP there are two levels, we are referring to two different evolutionary algorithms, one for upper and lower levels, respectively. Figure 9.10 represents the general idea of this approach.

One attempt to solve bilevel optimisation problems with co-evolutionary method was proposed by [55], named BiGA (bilevel genetic algorithm). The algorithm solved two optimisation problems iteratively. One was optimising the leader (upper) level problem for all the x variables and a subset of the y variables associated with the optimal basis of the follower's (lower) level problem. The other was optimising the follower problem with all the x variables fixed. In order to explore the optimal basis of the follower's problem, x was fixed, and then the corresponding 'optimal' y variables were returned to the leader problem. The basic genetic algorithm and its full operators is used in a dual population environment to solve both leader and follower problems. The algorithm was tested on four test functions, and the results showed that the BiGA approach was robust for solving different classes of BOP with reasonable performance. Recently, Legillon et al. [56] presented CoBRA, a parallel co-evolutionary algorithm for bilevel optimisation. Extending on the idea behind BiGA algorithm, it is a co-evolutionary metaheuristic algorithm consisting of two improving subpopulations, each corresponding to one level and periodically exchanging information with the other. The CoBRA approach was applied to solve a bilevel transportation problem, and the results found were better than those found by a classical hierarchical approach.

9.7 Applications

Bilevel optimisation is used in a wide area of applications. In structural optimisation or optimal shape design, the formulation of bilevel problems is very common. More specifically, it is common that the minimisation of the weight or cost of a structure

can be considered as an upper level objective with the decision variables such as the shape of the structure, choice of materials, amount of material, etc. being the lower level [2]. In [57], the minimisation of the final mass of the structure is defined as the upper level and the location of the nodes and struts as the lower level. In aerospace engineering, an example of bilevel formulation of a problem can be found in [58], where a bilevel optimisation strategy for wing design was developed. Optimisations of the wing planform and wing airfoil shapes were decoupled from each other, formulating a bilevel optimisation problem. In [59] the formulation of the truss topology problem with additional constraints on the displacements was approached as a bilevel problem. There, satisfying the displacement constraint was defined as the upper level problem and the minimisation of the compliance as the lower level problem. Herskovits et al. [60] formulated the problem of shape optimisation of nonlinear elastic solids in contact, as a bilevel problem, optimising simultaneously the shape and the nonlinear contact analysis, by taking the cost function minimisation with respect to the design variables as the upper level and, the minimisation of the elastic energy.

In a completely different research area, a very interesting application was proposed by Sinha et al. [61], where the parameter tuning of optimisation algorithms was presented as a bilevel problem. They tested this approach on two commonly used optimisation algorithms, differential evolution and Nelder-Mead, and they found that the approach converges towards the efficient parameters. This is very promising, as a suitable choice of parameters can have an important impact on the efficiency of the algorithm. More research in this area is needed though.

Other areas where bilevel optimisation is applied can be found in transport, for solving the toll setting problem, in the chemical industry, in environmental economics, in seller-buyer strategies, in optimal design, etc. A recent list of areas of application can be found in [5].

9.8 Summary

This chapter is a short introduction to multilevel and especially to bilevel optimisation problems. For a better understanding, a bilevel linear example is presented and compared to a biobjective equivalent. Moreover, special cases of bilevel problems are introduced, namely, multiobjective, multileader/multifollower and bilevel optimisation, under uncertainty problems along with minmax (worst-case scenario) as bilevel optimisation problem. The main solution algorithms categories, the classical and metaheuristic approaches are noted. More focus is given to metaheuristic approaches, providing a list of the algorithms that can be found in the literature. Last but not the least, some applications that have been formulated and solved as bilevel problems are shortly mentioned.

Acknowledgments This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant agreement number 722734, and through the SYNERGY Twinning project, H2020-TWINN-2015, Grant agreement number 692286.

References

1. K. Deb, D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms* (Wiley, New York, NY, 2001)
2. A. Sinha, Z. Lu, K. Deb, P. Malo, Bilevel Optimization based on Iterative Approximation of Multiple Mappings. Feb. 2017 [Online]. arXiv:1702.03394 [math]. Available <http://arxiv.org/abs/1702.03394>
3. J. Bracken, J.T. McGill, Mathematical programs with optimization problems in the constraints. *Oper. Res.* **21**(1), 37–44 (1973)
4. W. Candler, R. Norton, *Multi-Level Programming and Development Policy* (The World Bank, Washington, DC, 1977)
5. A. Sinha, P. Malo, K. Deb, A review on bilevel optimization: from classical to evolutionary approaches and applications. May 2017 [Online]. arXiv:1705.06270 [cs, math]. Available <http://arxiv.org/abs/1705.06270>
6. A. Valejo, M.C. Ferreira de Oliveira, G.P.R. Filho, A.D.A. Lopes, Multilevel approach for combinatorial optimization in bipartite network. *Knowledge-Based Systems*, March 2018 [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0950705118301539>
7. W.W. Hager (ed.), *Multiscale Optimization Methods and Applications*. Nonconvex optimization and its applications, vol. 82 (Springer, New York, 2006)
8. P. Korošec, J. Šilc, B. Robič, Solving the mesh-partitioning problem with an ant-colony algorithm, in *Parallel Computing*, vol. 30(5–6), pp. 785–801, May 2004 [Online]. Available <http://linkinghub.elsevier.com/retrieve/pii/S0167819104000432>
9. M.G. Fernández-Godino, C. Park, N.-H. Kim, R.T. Haftka, Review of multi-fidelity models (2016). Preprint. arXiv:1609.07196
10. J.A. Monschke, M.S. Eldred, Multilevel-Multifidelity Acceleration of PDE-Constrained Optimization. American Institute of Aeronautics and Astronautics, January 2017 [Online]. Available <http://arc.aiaa.org/doi/10.2514/6.2017-0132>
11. I.C. Kampolis, K.C. Giannakoglou, A multilevel approach to single- and multiobjective aerodynamic optimization. *Comput. Methods Appl. Mech. Eng.* **197**(33), 2963–2975 (2008) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0045782508000388>
12. C. Walshaw, Multilevel refinement for combinatorial optimisation problems. *Ann. Oper. Res.* **131**(1–4), 325–372. October 2004 [Online]. Available <https://link.springer.com/article/10.1023/B:ANOR.0000039525.80601.15>
13. A. Migdalas, P.M. Pardalos, P. Värbrand, *Multilevel Optimization: Algorithms and Applications* (Springer Science & Business Media, London, Heidelberg, 2013). google-Books-ID: 5pXaB-wAAQBAJ
14. T. Dudás, B. Klinz, G.J. Woeginger, The computational complexity of multi-level bottleneck programming problems, in *Multilevel Optimization: Algorithms and Applications* (Springer, Boston, MA, 1998), pp. 165–179
15. H.V. Stackelberg, Theory of the market economy (1952) [Online]. Available <http://agris.fao.org/agris-search/search.do?recordID=US201300604530>
16. S. Dempe, *Foundations of Bilevel Programming* (Springer Science & Business Media, Berlin/Heidelberg, 2002). google-Books-ID: 1sCWQ1MLMeQC
17. J. Lu, J. Han, Y. Hu, G. Zhang, Multilevel decision-making: a survey, in *Information Sciences*, vol. 346–347 (2016), pp. 463–487 [Online]. Available <http://linkinghub.elsevier.com/retrieve/pii/S0020025516300202>

18. P. Hansen, B. Jaumard, G. Savard, New branch-and-bound rules for linear bilevel programming. *SIAM J. Sci. Stat. Comput.* **13**(5), 1194–1217 (1992) [Online]. Available <https://epubs.siam.org/doi/abs/10.1137/0913069>
19. A. Sinha, Tutorial on Bilevel Optimization - Ankur Sinha - Aalto University Wiki, 2014 [Online]. Available <https://wiki.aalto.fi/display/~ansinha@aalto.fi>
20. S. Ruuska, K. Miettinen, M.M. Wiecek, Connections between single-level and bilevel multiobjective optimization. *J. Optim. Theory Appl.* **153**(1), 60–74 (2012) [Online]. Available <http://link.springer.com/10.1007/s10957-011-9943-y>
21. G. Ünlü, A linear bilevel programming algorithm based on bicriteria programming. *Comput. Oper. Res.* **14**(2), 173–179 (1987)
22. E.-G. Talbi (ed.), *Metaheuristics for Bi-level Optimization*. Studies in Computational Intelligence (Springer, Berlin/Heidelberg, 2013) [Online]. Available [//www.springer.com/gp/book/9783642378379](http://www.springer.com/gp/book/9783642378379)
23. G. Eichfelder, Multiobjective bilevel optimization. *Math. Program.* **123**(2), 419–449 (2010)
24. K. Deb, A. Sinha, Solving bilevel multi-objective optimization problems using evolutionary algorithms, in *International Conference on Evolutionary Multi-Criterion Optimization* (Springer, Heidelberg, 2009), pp. 110–124
25. Y. Yin, Multiobjective bilevel optimization for transportation planning and management problems. *J. Adv. Transp.* **36**(1), 93–105 (2002)
26. E. Roghanian, S.J. Sadjadi, M.-B. Aryanezhad, A probabilistic bi-level linear multi-objective programming problem to supply chain planning. *Appl. Math. Comput.* **188**(1), 786–800 (2007)
27. Y.G.A. Guangquan Zhang, J. Lu, *Multi-Level Decision Making: Models, Methods and Applications*. Intelligent Systems Reference Library, vol. 82 (Springer, Berlin/Heidelberg, 2015)
28. J. Lu, C. Shi, G. Zhang, On bilevel multi-follower decision making: general framework and solutions. *Inf. Sci.* **176**(11), 1607–1627 (2006)
29. K. Lachhwani, A. Dwivedi, Bi-level and multi-level programming problems: taxonomy of literature review and research issues. *Arch. Comput. Methods Eng.* **25**, 1–31 (2017)
30. F. Lin, X. Zhou, X. Lü, W. Song, Novel pre-pushing scheme for peer-assisted streaming network based on multi-leader multi-follower Stackelberg model. *Wirel. Pers. Commun.* **80**(1), 289–301 (2015)
31. Z. Lu, K. Deb, A. Sinha, Handling decision variable uncertainty in bilevel optimization problems. *IEEE*, May 2015, pp. 1683–1690 [Online]. Available <http://ieeexplore.ieee.org/document/7257089/>
32. A. Sinha, P. Malo, K. Deb, P. Korhonen, J. Wallenius, Solving bilevel multicriterion optimization problems with lower level decision uncertainty. *IEEE Trans. Evol. Comput.* **20**(2), 199–217 (2016)
33. B.L. Gorissen, İ. Yanıkoğlu, D. den Hertog, A practical guide to robust optimization. *Omega* **53**, 124–137 (2015)
34. A. Mucherino, M. Fuchs, X. Vasseur, S. Gratton, Variable neighborhood search for robust optimization and applications to aerodynamics, in *Large-Scale Scientific Computing, LSSC'11*. (Springer, Berlin, Heidelberg, 2012), pp. 230–237
35. I. Averbakh, Minmax regret solutions for minimax optimization problems with uncertainty. *Oper. Res. Lett.* **27**(2), 57–65 (2000)
36. D.-Z. Du, P.M. Pardalos, *Minimax and Applications*, vol. 4 (Springer Science & Business Media, Berlin, 2013)
37. M. Vasile, On the solution of min-max problems in robust optimization, in *The EVOLVE 2014 International Conference, A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computing*, Jian-Guo Hotel, July 2014 [Online]. Available <https://strathprints.strath.ac.uk/52249/>
38. C. Ortega, M. Vasile, New heuristics for multi-objective worst-case optimization in evidence-based robust design, in *2017 IEEE Congress on Evolutionary Computation (CEC)*, June 2017, pp. 1519–1526
39. R.I. Lung, D. Dumitrescu, A new evolutionary approach to minimax problems, in *2011 IEEE Congress of Evolutionary Computation (CEC)*, June 2011, pp. 1902–1905

40. A. Zhou, Q. Zhang, A surrogate-assisted evolutionary algorithm for minimax optimization, in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–7
41. C.D. Kolstad, L.S. Lasdon, Derivative evaluation and computational experience with large bilevel mathematical programs. *J. Optim. Theory Appl.* **65**(3), 485–499 (1990) [Online]. Available <http://link.springer.com/10.1007/BF00939562>
42. E. Aiyoshi, K. Shimizu, Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Trans. Syst. Man Cybern.* **SMC-11**(6), 444–449 (1981) [Online]. Available <https://keio.pure.elsevier.com/ja/publications/hierarchical-decentralized-systems-and-its-new-solution-by-a-barr>
43. H. Jiye, L. Guoshan, W. Shouyang, A new descent algorithm for solving quadratic bilevel programming problems. *Acta Math. Appl. Sin.* **16**(3), 235–244 (2000). [Online]. Available <http://link.springer.com/10.1007/BF02679888>
44. S. Dempe, J. Dutta, S. Lohse, Optimality conditions for bilevel programming problems. *Optimization* **55**(5–6), 505–524. October 2006 [Online]. Available <http://www.tandfonline.com/doi/abs/10.1080/02331930600816189>
45. A. Sinha, P. Malo, K. Deb, Evolutionary algorithm for bilevel optimization using approximations of the lower level optimal solution mapping. *Eur. J. Oper. Res.* **257**(2), 395–411 (2017) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0377221716306634>
46. R. Mathieu, L. Pittard, G. Anandalingam, Genetic algorithm based approach to bi-level linear programming. *RAIRO - Oper. Res.* **28**(1), 1–21 (1994) [Online]. Available <http://www.rairo-ro.org/10.1051/ro/1994280100011>
47. Y. Yafeng, Genetic-algorithms-based approach for bilevel programming models. *J. Transp. Eng.* **126**(2), 115–120 (2000) [Online]. Available [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-947X\(2000\)126:2\(115\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-947X(2000)126:2(115))
48. H.W. Kuhn, A.W. Tucker, Nonlinear programming, in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950* (University of California Press, Berkeley/Los Angeles, 1951), pp. 481–492
49. Y. Wang, Y.-C. Jiao, H. Li, An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **35**(2), 221–232 (2005) [Online]. Available <http://ieeexplore.ieee.org/document/1424196/>
50. Y. Wang, H. Li, C. Dang, A new evolutionary algorithm for a class of nonlinear bilevel programming problems and its global convergence. *INFORMS J. Comput.* **23**(4), 618–629 (2011) [Online]. Available <http://pubsonline.informs.org/doi/abs/10.1287/ijoc.1100.0430>
51. M.M. Islam, H.K. Singh, T. Ray, A surrogate assisted approach for single-objective bilevel optimization. *IEEE Trans. Evol. Comput.* **21**(5), 681–696 (2017)
52. A. Sinha, P. Malo, K. Deb, Solving optimistic bilevel programs by iteratively approximating lower level optimal value function. *IEEE*, July 2016, pp. 1877–1884 [Online]. Available <http://ieeexplore.ieee.org/document/7744017/>
53. S. Ankur, M. Pekka, D. Kalyanmoy, Efficient evolutionary algorithm for single-objective bilevel optimization. March 2013 [Online]. arXiv:1303.3901 [cs]. Available <http://arxiv.org/abs/1303.3901>
54. J.S. Angelo, E. Krempser, H.J.C. Barbosa, Differential evolution assisted by a surrogate model for bilevel programming problems, in *2014 IEEE Congress on Evolutionary Computation (CEC)*, July 2014, pp. 1784–1791
55. V. Oduguwa, R. Roy, Bi-level optimisation using genetic algorithm. *IEEE Comput. Soc* 322–327 (2002) [Online]. Available: <http://ieeexplore.ieee.org/document/1048121/>
56. F. Legillon, A. Liefoghe, E.-G. Talbi, Cobra: a coevolutionary metaheuristic for bi-level optimization, in *Metaheuristics for Bi-level Optimization*. Studies in Computational Intelligence (Springer, Berlin, Heidelberg, 2013), pp. 95–114
57. L.U. Hansen, P. Horst, Multilevel optimization in aircraft structural design evaluation. *Comput. Struct.* **86**(1), 104–118 (2008) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0045794907001952>

58. A. Elham, M.J.L. van Tooren, J. Sobieszczanski-Sobieski, Bilevel optimization strategy for aircraft wing design using parallel computing. *AIAA J.* **52**(8), 1770–1783 (2014) [Online]. Available <http://arc.aiaa.org/doi/10.2514/1.J052696>
59. M. Kočvara, Topology optimization with displacement constraints: a bilevel programming approach. *Struct. Optim.* **14**(4), 256–263 (1997) [Online]. Available <https://link.springer.com/article/10.1007/BF01197948>
60. J. Herskovits, A. Leontiev, G. Dias, G. Santos, Contact shape optimization: a bilevel programming approach. *Struct. Multidiscipl. Optim.* **20**(3), 214–221 (2000) [Online]. Available <http://link.springer.com/10.1007/s001580050149>
61. A. Sinha, P. Malo, P. Xu, K. Deb, A bilevel optimization approach to automated parameter tuning, in *GECCO '14: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (ACM Press, New York, NY, 2014), pp. 847–854 [Online]. Available <http://dl.acm.org/citation.cfm?doid=2576768.2598221>

Chapter 10

Sequential Parameter Optimization for Mixed-Discrete Problems



Lorenzo Gentile, Thomas Bartz-Beielstein, and Martin Zaefferer

Abstract Mixed-discrete optimization deals with mathematical optimization problems with multiple types of variables: discrete (nominal) taking values from a not-sortable set of possible elements, integer variables and variables taking values in a continuous domain. Mixed-discrete problems appear naturally in many contexts such as in the real world in the engineering domain, bioinformatics and data sciences, and this has led to an increased interest in the design of strong algorithms for different variants of the problem. Much effort has been spent over the last decades in studying and developing new methodologies, but unfortunately mixed-discrete optimization problems are much less understood than their “non-mixed” counterparts. In this chapter we will focus on the rather new approaches to handle mixed-discrete problems by means of surrogate methods.

Keywords Optimization · Mixed-discrete optimization · Sequential parameter optimisation

10.1 Introduction

Many real-world optimization problems consider the optimization of ordinal integers, categorical integers, binary variables, permutations, strings, trees or graphs structures in general. These real-world problems pose complex search spaces which require a deep understanding of the underlying solution representations. Some of them, for example, integers, are more suitable to be treated by classic optimization algorithms. Others, such as trees, have to be handled by specifically developed optimization algorithms. In general, solving these kinds of problems usually necessitates a significant number of objective function evaluations. However, in many engineering problems, a single evaluation is based either on experimental or

L. Gentile (✉) · T. Bartz-Beielstein · M. Zaefferer
Institute for Data Science, Engineering, and Analytics, TH Köln, Köln, Germany
e-mail: lorenzo.gentile@th-koeln.de; thomas.bartz-beielstein@th-koeln.de;
martin.zaefferer@th-koeln.de

numerical analysis. This causes significant costs with respect to time or resources. Surrogate model-based optimization (SMBO) aims to handle the complex variable structures and the limited budget simultaneously. Sequential parameter optimization (SPO) pursues the identification of global optima, making advantage of a budget allocation process that maximizes the information gaining in promising regions. This chapter aims to show an efficient method to face mixed-discrete optimization problems utilizing SPO. Particularly, the chapter is structured as follows: Sect. 10.2 introduces the problem definition, Sect. 10.3 describes the challenges that are common in this problem domain, Sect. 10.4 contains a thorough description of SPO, and finally an application of SPO on real-world discrete-mixed optimization problem is presented in Sect. 10.5.

10.2 Problem Definition

Optimization can be seen as the process of searching for the best candidate solution in the search space, which maximizes or minimizes an objective function. Without loss of generality, we refer to optimization as a minimization process. In this chapter, we will focus on describing problems including real-valued variables, ordinal integers, and categorical (uncountable) variables.

Let $f : \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \times \mathbb{D}^{n_d} \rightarrow \mathbb{R}$ denote the objective function to be optimized, $g_j : \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \times \mathbb{D}^{n_d} \rightarrow \mathbb{R}$, $1 \leq j \leq n_g$ the inequality constraints and $h_k : \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \times \mathbb{D}^{n_d} \rightarrow \mathbb{R}$, $1 \leq k \leq n_h$ the equality constraints.

The problem of mixed-discrete optimization can be formalized as follows:

$$\begin{aligned} & \min_x f(x) \quad \text{where } x \in \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z} \times \mathbb{D}^{n_d} \\ & \text{subject to } \begin{cases} g_j \leq 0, 1 \leq j \leq n_g \\ h_k = 0, 1 \leq k \leq n_h \\ r_i \in [r_i^{\min}, r_i^{\max}], 1 \leq i \leq n_r \\ z_i \in [z_i^{\min}, z_i^{\max}], 1 \leq i \leq n_z \\ d_i = \{d_i^1, \dots, d_i^{N_i}\}, 1 \leq i \leq n_d \end{cases} \end{aligned} \quad (10.1)$$

where r_i^{\min} and r_i^{\max} define the lower and upper bounds that the n_r real variables r_i can assume, z_i^{\min} and z_i^{\max} define the lower and upper bounds that the n_z integer variables z_i can assume, d_i is the set of the possible values that the i -th discrete variable can assume and finally n_d is the number of discrete variables. The input variables will be referred to design variables.

10.3 Challenges in Real-World Optimization

10.3.1 Problem Features

Over the years, a large number of optimization methods have been proposed, and new algorithms are developed every day to improve their general performance. However, it has been stated by Wolpert and Macready [57] that any algorithm's improved performance over one class of problems is offset by a performance loss over another class. Hence, the identification of problem features becomes a crucial stage in the development and selection of optimization algorithms. Among all potential problem features, the ones that mostly affect the performances in mixed-discrete optimization can be listed as follows: high dimensionality, uncertainties, computationally expensive evaluations, complex landscapes and black-box problems [55].

Here, "black-box" implies that no knowledge about the function is available and any knowledge can only be derived by evaluating the function itself. It is often impossible to predict the response of the function because the physical phenomena are not fully understood or the modeling strategy leads to bias and undesired, unknown sensitivities. The integration of optimization methodologies with computational analysis and simulations is of some importance in this context. This lack of knowledge is especially problematic when function evaluations are expensive: Black-box optimization processes inherently require numerous evaluations of objective functions [28]. Therefore, although numerous non-gradient optimization methods are available for cheap black-box functions, more sophisticated methods are necessary to deal with limited evaluation budgets.

Determining whether the best solution currently known is a local or a global optimum is often difficult. This issue typically arises for multi-modal problems (if the function has multiple maxima and minima). Moreover, difficult fitness landscapes may exhibit *deceptiveness* [10]. Deceptive objective functions can trap the optimizer by a large basin of attraction, which leads the search process away from the global optimum in favour of a local one.

If an area with a better average fitness compared to other regions has been found, the optimization algorithm will consider it as promising and will focus on the exploration of this region. This assumes that such areas are likely to contain the true optimum. Hence, developing an algorithm that is able to interpret the function response correctly is a demanding task.

In many cases, this problem can be solved by choosing the correct optimization strategy and performing a preliminary algorithm tuning. For example, a population-based optimizer's ability to distinguish the global optimum from a local optimum often relies on the chosen population size. Moreover, maintaining diversity in the population helps to avoid premature convergence [55]. It is also clear that the optimality of the algorithms' parameters changes during the optimization process in case of multi-modal problems: In the beginning, algorithms should be more explorative. This leads to the fastest identification of all the promising areas and

would help escaping misleading local minima. On the contrary, at the end of the process, exploitation would gain more importance, assuming that the most promising area has already been identified. For these reasons, algorithms able to auto-tune (i.e. perform on-the-fly parameter control) all along the optimization process, such as [31, 40], can be a promising choice. More details on auto-tune can be found in Chap. 11. The combination of these features strongly increases the problem difficulty. Optimization algorithms need to be designed to solve specific problems, presenting different combinations of these features. We will focus on two essential problems in this chapter, namely, the dimensionality and the uncertainty.

10.3.2 High Dimensionality

The dimensionality of the search space is defined by the number of design variables. Referring to the notation in Eq. (10.1), we define the dimensionality as $n = n_r + n_z + n_d$. It is intuitive that a large number of variables pose a demanding challenge that affects many algorithm's aspects. Dealing with this particular problem requires a great modeling capability, a huge amount of acquired data and, consequently, a large budget of objective function evaluations. Every modeling technique requires a sufficiently large dataset such that an accurate model can be trained.

Furthermore, high dimensionality leads to severe practical issues in the development of surrogate models as well. Kriging, besides linear regression, is one of the most popular techniques in SMBO, see [8]. For example, depending on the employed distance measures [1], it is widely recognized that Kriging may perform poorly for problems with more than approximately 20 variables [19].

A spectrum of countermeasures to these issues comes from different fields of engineering and data analytics. Most commonly, methods attempt to use some screening or mapping approach. The former attempts to remove insignificant variables, while the latter attempts to map the original search space to a low-dimensional subspace.

10.3.2.1 Screening

In the effort of reducing the problem complexity and dimensionality, screening identifies and retains important input variables and interaction terms. Screening is often implemented via sampling and the analysis of sampling results [46].

Sensitivity analysis studies how the variability of a function's output responds to changes of its inputs. It includes local and global sensitivity analyses. The local sensitivity indicates the variability of the output with respect to input variable changes at a given point; in other words it evaluates the numeric partial derivatives. The global sensitivity, contrarily, explains the global variability of the output over the entire design space, which provides an overall view of the impact of input

variables on the output. One example of sensitivity analysis applied to aircraft design is given in [48].

A classic method for screening and sensitivity analysis using experimental designs is the modeling and analysis of regression models [14]. Common examples are the analysis of p -values in linear regression, mean decrease impurity in random forests [12] and the theta values of Kriging models [19].

For example: We consider a simplified variant of the optimization problem introduced in Eq. (10.1). The input variables are real-valued, and no equality and inequality constraints are imposed. In this situation we obtain the following optimization problem:

$$\min_{\vec{x} \in \mathbb{R}^2} f(\vec{x})$$

with $\vec{x} = (x_1, x_2)$ we obtain the linear regression model

$$\hat{f}(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

The coefficients β_1 and β_2 are interpreted as the estimated change in the objective function corresponding to one unit change in a variable, when all the other variables are held constant [36]. The p -values [36] for the coefficients indicate whether these relationships are statistically significant. Intuitively, it is possible to set a threshold value for the p -values over which the corresponding variables are considered uncorrelated to the objective function and, thus, can be neglected. In the scope of linear regression, screening designs have been developed, which are applicable in the context of small function evaluation budgets [52].

A model-independent approach for variable screening that varies one factor at a time has been proposed by Morris [38]. This method, unlike the stepwise variable selection [24], requires a number of model evaluations that are linearly dependent by the number of design variables. This strategy aims to estimate the overall effect of the variables and the ensemble of the second-order and higher-order effects addressing two sensitivity measures per design variable [43].

Screening processes can be also directly integrated in the metamodel building. A cross-validated moving least squares approach in which one variable of the problem design represented the screening has been proposed in [52]. However, there is the risk of reducing accuracy due to the omitted dimensions.

10.3.2.2 Mapping

Mapping has a broad meaning including projection, nonlinear mapping and parameter space transformation. A mapping procedure transforms a set of correlated variables into a smaller set of new uncorrelated ones that retain most of the original information. One popular approach that relies on linear analyses is the principal

component analysis (PCA). This method is especially used for problems with only continuous variables [13].

Contrarily, other methods are based on nonlinear mapping and projection. Space mapping (SM) intends to map the design space of a “coarse” and low-dimensional model to a fine, “expensive” and higher-dimensional one [5]. A good survey of related approaches is given in [46].

Another class of dimensionality reduction approaches is based on unsupervised learning. Two promising examples are *autoencoders* and *self-organizing maps*. Autoencoders are neural networks that aim to reconstruct their own inputs. As such, an encoder network maps from the high-dimensional space to the coded space. Then, a decoder network maps back to the high-dimensional space, with as little loss of information as possible. By constraining the coded space to have a smaller dimension than the input space, the autoencoder is forced to learn the most salient features of the input data [42]. Self-organizing maps are a particularly interesting class of unsupervised systems that are based on competitive learning. The output neurons compete among themselves to be activated. Hence, only one is activated at one particular time. This competitive system forces the neurons to organize themselves. Commonly, self-organizing maps target to map from the high-dimensional space to one- or two-dimensional space [29].

10.3.3 Uncertainty

A large variety of optimization problems in scheduling, finance, transportation and engineering design requires that decisions are made in the presence of uncertainty. Uncertainties are present in all real-world application problems, e.g. due to inaccuracies in the manufacturing process, uncertain operating conditions or system component failures. However, different forms of uncertainties can be distinguished; a good overview can be found in [26]. In the following a description of the most common forms of uncertainty in real-world application problems, *noise* and *robustness*, will be given.

An optimization problem is considered subjected to noise if the objective function is perturbed. This can be due to several factors such as sensor measurements errors or heuristic simulations. Mathematically, noise is often assumed normally distributed with zero mean and variance [26].

In other cases, perturbation can afflict the design variables. Therefore, it is often required that the optimal solution should still work satisfactorily when the design variables change slightly, e.g. due to manufacturing tolerances [26].

In these cases, successfully performing global optimization means facing a variety of challenging issues. Resources have to be allocated to perform an uncertainty analysis in order to direct the research to stable and robust optima.

The correct balance between exploitation, exploration and uncertainty quantification has to be addressed. Furthermore, adopting surrogate model-based optimization in noisy functions causes an additional problem. The use of derivative-based

optimization techniques can lead to regions with seemingly good function caused by a misinterpretation of the noisy data. This issue appears if surrogate models confuse noise with the actual behaviour of the objective function.

Thus, finding efficient methods to deal with uncertainty appears to be a non-trivial problem. A popular approach is searching for *robust* optima. This is done by replacing the deterministic objective function f in favour of a modified \tilde{f} that feeds an estimation of f back to the optimizer. This is evaluated observing the response of f a number of times with the same design solution. Popular examples of \tilde{f} are the *expected value*, *expected value + k standard deviation* (the importance of the *standard deviations* in respect of the *mean* is weighted by the coefficient k) or the *95% quantile*. The obvious drawback of this method is the considerable number of repetitions of f that are needed to make an accurate prediction of the *robust* objective function. However, researchers are proposing methods to mitigate the computational effort in estimating \tilde{f} replacing classic uncertainty quantification methods, such as quasi-Monte Carlo quadrature, in favour of polynomial chaos with coefficients determined by sparse quadrature and by point collocation, radial basis function or Kriging models [32].

10.4 Sequential Parameter Optimization

The sequential parameter optimization toolbox (SPOT) [6] is an optimization framework which is based on surrogate model-based optimization. The aim of SMBO is to train a cheap numerical model that approximates the objective function and utilize it to reduce the computational effort.

Initially proposed for algorithm tuning of metaheuristics, SPOT is a sophisticated tool capable of handling both continuous and mixed-discrete problems [8]. SPOT spends the available budget in a sequential manner to maximize information gain and is particularly efficient for expensive problems. SPOT finds improved solutions in the following way (see Algorithm 1): First, the search space is sampled with an experimental design plan (see Sect. 10.4.1). With these samples, a first surrogate model is constructed. Then, an infill criterion is optimized on the surrogate to find new promising candidate solutions. The suggested candidates are evaluated with the real objective function, and the surrogate is updated with the observed information. In the following we will describe the fundamental steps of the SPOT methodology. A more detailed description of SPOT can be found in [9].

Algorithm 1 Sequential parameter optimization

```

1:  $t = 0$ .
2: Initialize a number of  $k$  samples  $P_t = \{x_i, 1 \leq i \leq k\}$ .
3: Select a suitable surrogate model  $M_t(x)$ .
4: Evaluate  $P_t$  on  $f$  to get observations  $O_t = \{(x_i, y_i), 1 \leq i \leq k\}$ .
5: while not Termination Criterion do
6:   Build a model  $M_t$  with  $O_t$ 
7:   Optimize  $M_t$  infill criteria to get  $x_t^*$ 
8:   Evaluate  $x_t^*$  on  $f$  to get  $y_t^*$ 
9:   Update dataset  $O_{t+1} = \{O_t, (x_t^*, y_t^*)\}$ .
10:   $t = t + 1$ .
11: end while

```

10.4.1 Initial Design

10.4.1.1 Strategies for Design of Experiment

The first step of SPOT (see Algorithm 1) is the determination of the initial dataset that will be used to train the first surrogate model. In order to build a moderately accurate model, the initial design should cover, if possible, the complete feasible search space. To that end, we rely on sampling methods.

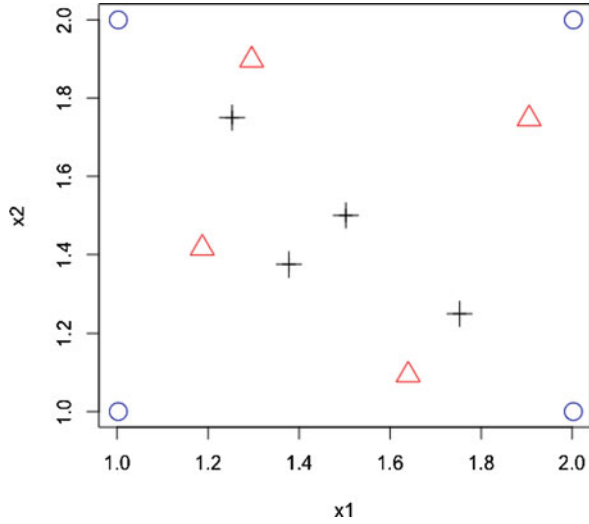
Sampling methods can be classified as *deterministic* and *stochastic* methods. Examples of pure deterministic sampling are grid designs, full factorial designs and Sobol sequences [49]. Stochastic methods try to create unbiased subsets of the original search space. They often optimize a certain criterion such as D-optimality or I-optimality [36]. This class includes basic random sampling, stratified sampling (e.g. Latin hypercube sampling) and fractional factorial designs.

10.4.1.2 Latin Hypercube Sampling

As a representative of stratified sampling, one of the most commonly employed methods is Latin hypercube sampling (LHS) [34]. LHS creates multidimensional designs. Given the number of samples n , all $n_r + n_z + n_d$ dimensions are divided into n intervals. LHS samples a point from each stratum. Different variants for choosing a point in each stratum exist. For example, median LHS uses the median value of each interval, while random LHS selects a random point within each interval.

This procedure has to be adapted to also treat categorical and discrete variables. One of the simplest solutions for ordinal variables is assuming that all variables are continuous and then using floor, ceiling or rounding operations. Dummy variables may be employed for categorical parameters. Or else, categorical parameters may be mapped to ordinal integers.

Fig. 10.1 Example of three different sampling methods in creating an initial design of size 4 in a two-dimensional space: Sobol sequence (red triangles), Latin hypercube (black crosses) and full factorial design (blue circles)



10.4.1.3 Factorial Designs

In the field of design of experiments, a set of statistically well-profound designs have emerged, which are commonly applied to analyse and optimize industrial problems. Common designs of this field are full factorial, fractional factorial, Box-Behnken and central composite [36]. All these designs are designed to fit linear models for the response surface methodology, commonly with second-order and quadratic effects. For example, in a full factorial design for two variables with two levels for each design variable, a set of 2^2 evenly spaced points is determined. In case of continuous variables, they are determined by $[r^{\min}, r^{\max}]$. With this design, we are able to analyse main and second-order effects. For quadratic effects, centre points need to be added. A full factorial design has the disadvantage of requiring an exponentially increasing number of experiments with rising number of variables. To prohibit an infeasible number of experiments, usually optimized fractional factorial or other screening designs are utilized.

From Fig. 10.1, one can see that, contrary to LHS and the Sobol sequence, a full factorial sampling exhibits a particular grid structure that eases distinguishing the effects of all design variables on the objective function.

10.4.2 Modeling

10.4.2.1 Modeling in Mixed-Integer Space

Once the first dataset has been created and observed, it is used to train a surrogate model that aims to replicate the behaviour of the objective function (see Algo-

rithm 1). In SPOT, a surrogate model is used to determine promising candidate solutions. To that end, it aims to learn the relation between problem variables and the corresponding function's response.

Compared to their frequent use for real-valued problems, surrogate model-driven approaches are less often used in mixed optimization [27]. According to Bartz-Beielstein and Zaefferer [8], few expensive, real-world optimization problems of this type have been brought to the science community's attention, e.g. in the engineering domain [3, 50, 51, 54], bioinformatics [41] or data science [47]. One reason for the scarce use of discrete surrogate model-based optimization is the availability of suitable methods. Bartz-Beielstein and Zaefferer [8] identified six strategies for surrogate modeling in mixed-discrete search spaces: the naive approach, customized models, inherently discrete models, feature extraction, mapping and similarity-based models. These strategies explain how modeling techniques can be used in the general cases. The six strategies are not mutually exclusive. Some methods may belong to several categories or combine different strategies. Here, we focus on three of the six strategies: the naive approach, inherently discrete models and similarity-based models since these are more commonly used.

10.4.2.2 The Naive Approach

The naive approach to discrete modeling is to ignore the discrete nature of the search space. Standard continuous methods are applied to solve the optimization problem. An application of this approach can be found in [7]. There, the authors faced an expensive parameter tuning problem and employed Kriging models. Especially if the discrete variables are of an ordinal nature, the naive approach may be successful. Indeed, this strategy could even be adopted to deal with categorical variables: Addressing an arbitrary order would create a one-to-one correlation between categorical variables and ordinal values. Several potential drawbacks can arise if this strategy is employed for problems that are too complex:

- Large areas of redundancy in the model's input space
- Creation of infeasible solutions
- Degeneration of performance due to bias caused by a misinterpretation of variables

The naive approach is attractive, due to its ease of use and the ability to stick to continuous variable handling methods. Practitioners have to carefully evaluate if this option suits the characteristics of the problem under study.

10.4.2.3 Inherently Discrete Models

There are models that are discrete in their design and hence need no further adaptation to discrete variables. For example, tree-based models, like regression trees or random forests, are inherently discrete models. A representative application

of this strategy can be found in [25], where an optimization process based on random forest models has been employed for a high-dimensional algorithm tuning problem. On the one hand, inherently discrete models are easy to use, since they require no additional work to adapt them to discrete problems. On the other hand, in a mixed-variable case, a tree-based model would not be well-suited to represent the continuous parameters in the mixture. Also, tree-based models may not provide the useful features that models like Kriging has. For example, uncertainty estimates can be derived from random forests, but unlike Kriging, these estimates do not go to zero at observed sample locations.

10.4.2.4 Similarity-Based Models

Similarity-based modeling is a promising strategy that is gaining more and more interest. Here, suitable measures of similarity are used to model discrete data. With respect to their interpretation and use, the measures are referred to as similarity measures, dissimilarity measures, distance measures, correlation measures or kernels. Although this approach is potentially very powerful, it requires the definition of proper measures. This may be problematic if these measures have to fulfil further requirements, e.g. definiteness, and if the problem involves different types of variables.

Fonseca et al. [18] defined similarity-based models that keep a memory of solutions and estimate the performance of new samples by comparing them to that memory. Three models from this class are of particular interest: radial basis function networks (RBFN), support vector machine (SVM) and Kriging. Various model-based variants applied to different optimization problems can be found in the literature, e.g. [4, 16, 21, 22, 25, 30, 37, 58]. Several of these works involve the development of appropriate similarity measures for discrete or mixed search spaces.

As these previous developments indicate, similarity-based models like Kriging are very promising approaches towards handling mixed and discrete variables. Hence, we focus on Kriging in the following.

10.4.2.5 Handling Factor Variables in Kriging Model

Kriging is a similarity-based model and assumes that the data follows a multi-variate Gaussian distribution, where errors are spatially correlated. A detailed and comprehensible description of Kriging is given by Forrester et al. [19].

We consider a simplified variant of the optimization problem defined in Eq. (10.1):

$$\min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$$

and no equality and inequality constraints are used. Importantly, the spatial correlation of the data is encoded within a kernel or correlation function. A frequently employed correlation function that models the correlation between samples (or candidate solutions) is the Gaussian kernel $k(x, x') = \exp(-\sum_{i=1}^n \theta_i |x_i - x'_i|^2)$. Here, n is the number of modeled variables (search space dimension) and θ_i is a parameter of the kernel (determined by maximum likelihood estimation (MLE)). Furthermore, x as well as x' are potential candidate solutions (or samples). Employing such a kernel, a Kriging model produces the following predictor:

$$\hat{y}(z^*) = \hat{\mu} + \mathbf{k}^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (10.2)$$

where \mathbf{y} are the training observations, $\hat{y}(z^*)$ is the predicted function value of a new sample z^* , $\hat{\mu}$ represents the process mean determined by MLE, $\mathbf{1}$ is a vector of ones, \mathbf{K} is the matrix that collects all pair-wise correlations of the training samples \mathbf{Z} and \mathbf{k} is the column vector of correlations between the set of training samples \mathbf{Z} and the new sample z^* . After appropriate training, such a predictor may be employed to replace an expensive objective function.

The success of Kriging in the field of real-world application problems mostly relies on the possibility to estimate the uncertainty of the predictor. This feature assumes a prominent role in applications in which the limited number of observations that can be performed inhibits an exhaustive exploration of the search space. In these cases, the estimate of the uncertainty can be used to balance exploration and exploitation by computing the expected improvement (EI) of candidate solutions [35]. The uncertainty of the model is computed with

$$\hat{\sigma}^2(z^*) = \sigma_{process}^2(1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k}), \quad (10.3)$$

where $\sigma_{process}^2$ is the process variance, determined by MLE. If the uncertainty is zero, the EI is also zero. Else, the uncertainty is non-zero, and the EI is

$$EI(z^*) = \mathbf{y}_{imp} \Phi\left(\frac{\mathbf{y}_{imp}}{\hat{\sigma}(z^*)}\right) + \hat{\sigma}(z^*) \phi\left(\frac{\mathbf{y}_{imp}}{\hat{\sigma}(z^*)}\right),$$

where $\mathbf{y}_{imp} = \min(\mathbf{y}) - \hat{y}(z^*)$. $\Phi()$ indicates the normal cumulative distribution function. Respectively, $\phi()$ is the probability density function.

It has to be noted that the above description of Kriging presents an interpolating model, which assumes zero error at already observed locations. Clearly, this does not take noise or uncertainty into account. One way to account for noise is to introduce the so-called nugget effect. This essentially adds a constant value η to the diagonal of the kernel matrix \mathbf{K} . The parameter η is determined by MLE. The nugget effect enables the model to regress the observed data and hence smoothens noisy observations.

Until now, we discussed Kriging in the context of real-valued search spaces. It is also applicable to “mixed” search space, where an appropriate kernel is available.

With respect to mixed or discrete problems, Kriging is actually very flexible. By changing the kernel (or correlation) function, any search space may be modeled with Kriging [8, 59]. The flexibility of this modeling method renders Kriging one of the most promising mixed-variable models. Take for example a typical problem characterization from algorithm tuning: Parameters like mutation rates may be real-valued, and the choice between different mutation operators may be a categorical parameter. Hence, if x_i (the i -th dimension of a parameter configuration \vec{x}) is a factor variable, Hamming distance can be used, otherwise the absolute deviation may be used for real-valued variables. The reader is referred to the discussion in [8].

10.4.3 Optimization Algorithms for the Metamodel

As shown in Algorithm 1, the next step in the SPOT methodology, after constructing a model, consists in the employment of optimizers that search for promising candidate solutions. Standard techniques from mathematical programming [56], the so-called mixed-integer nonlinear programming methods [17], are commonly not applicable to deterministic optimization of real-world application problems. These methods, such as outer approximation [15], branch and bound [11], and generalized Benders decomposition [20], have difficulties with the mixed design space, multimodality, uncertainty in the observations and unknown black-box properties. A consolidated alternative consists in the employment of metaheuristics for mixed-discrete optimization [31]. These strategies propose to heuristically determine solutions that improve the objective function value.

In cases where mathematical programming techniques are not flexible enough to yield satisfying results, heuristic search for solutions that improve the objective function value can lead to interesting results. Metaheuristics for mixed-discrete optimization are generally categorized in two classes:

- *Hierarchical approaches* solve problems with continuous variables together and discrete variables by considering the original optimization as a bi-level problem. The discrete variables are optimized by the upper level optimization process and the continuous parameters are optimized in the lower level [33, 53].
- *Simultaneous approaches* optimize discrete and continuous parameters simultaneously. In this approach, we consider that a similarity of parameter vectors due to an appropriate metric is equivalent to being positively correlated to the similarity in function values [31, 44].

In the following we will highlight the peculiarities of an algorithm from the class of simultaneous approaches. These algorithms are a better choice for our purposes. They need fewer observations and consider correlations between discrete and continuous variables. This is in contrast to the hierarchical approaches where variables of different type are strictly separated from each other. Particularly, we will discuss the mixed-integer evolution strategy (MIES) proposed in [31].

10.4.4 MIES

Evolution strategies (ES) are metaheuristics that follow the concept of natural evolution. An individual in an ES contains the information about one solution candidate. This individual is subject to recombination, mutation and selection operations. By evolving sets, or populations of individuals, the ES try to find improved solutions.

In MIES, an individual contains information about real-valued variables, ordinal integer variables and categorical variables. Parameters of the probability distribution used in the mutation operator (such as standard deviations or step sizes) are also stored in the individual for the purpose of self-adaptive parameter control. The latter parameters are referred to as *strategy parameters*. As a consequence the domain of an individual \mathbb{I} can be expressed as follows:

$$\mathbb{I} = R_1 \times \dots \times R_{n_r} \times Z_1 \times \dots \times Z_{n_z} \times D_1 \times \dots \times D_{n_d} \times A_s \quad (10.4)$$

with $A_s = \mathbb{R}_+^{n_\sigma + n_\zeta} \times [0, 1]^{n_p}$ being the domain of the *strategy variables*. Correspondingly, an individual of a population can be represented as

$$\vec{a} = (r_1, \dots, r_{n_r}, z_1, \dots, z_{n_z}, d_1, \dots, d_{n_d}, \sigma_1, \dots, \sigma_{n_\sigma}, \zeta_1, \dots, \zeta_{n_\zeta}, p_1, \dots, p_{n_p}) \quad (10.5)$$

The so-called *design variables* $r_1, \dots, r_{n_r}, z_1, \dots, z_{n_z}, d_1, \dots, d_{n_d}$ determine the objective function value and thus the fitness of the individual. The *strategy variables* $\sigma_1, \dots, \sigma_{n_\sigma}$ are standard deviations used in the mutation of the real-valued variables, and $\zeta_1, \dots, \zeta_{n_\zeta}$ denote mean step sizes in the mutation of the integer parameters. Finally, p_1, \dots, p_{n_p} denote mutation probabilities (or rates) for the nominal discrete object parameters. MIES is considered a self-adaptive process because the strength of the mutation parameters continuously evolves during the optimization. Hence, the mutation strength itself is also governed by an evolutionary process. The philosophy behind self-adaptation is that the evolutionary process can solve two problems simultaneously: the determination of the best strategy variables and the determination of the best object variables. More details on self-adaptation can be found in Chap. 11.

The first population $P(0)$ of μ individuals is generated by uniform random sampling from \mathbb{I} . Then, the main loop of the MIES algorithm starts. In a first step, the algorithm generates the set of λ new offspring individuals with the following procedure. Two parents are randomly selected from the population, and an offspring is generated by recombination and mutation. The recombination operator can be subdivided into two steps, selection of the parents and recombining the selected parents. The two parents $c_1, c_2 \in I$ are selected randomly, from the parental generation for each of the offspring individuals. In MIES, two different types of recombination are used: dominant and intermediate [45]. The first one is adopted for solution variables and consists of a random selection of one of the corresponding parental parameters for each offspring vector position. The latter is used for

recombining the strategy parameters and computes the mean of both parental vectors. The mutation of the offspring relies on operators acting differently on real, integer and discrete variables, all respecting the requirements for a mutation strategy in the search spaces: *accessibility*, *feasibility*, *symmetry*, *similarity*, *scalability*, and *maximal entropy* [31].

The MIES achieves this by adding normal distributed noise to real-valued variables. For integer variables, the distribution is based on the difference of two geometrical distributions. Categorical variables are simply re-sampled (uniform randomly) with some probability p [31].

In the next step of the iteration, the λ offspring individuals are ranked on the basis of the objective function. The μ best individuals out of the union of the λ offspring individuals and the μ parental individuals are selected. The generational loop is repeated until the number of evaluation exceeds the budget.

10.5 Case Study: Optimization of Composite Multi-Layered Plate

In this section, a mixed-discrete problem based on a real-world application is discussed. The problem consists in the design optimization of a composite multi-layered plate. Our idea is to demonstrate the difficulties that researchers and practitioners encounter when facing black-box, time-consuming, mixed-discrete problems under uncertainty through an illustrative example. A performance comparison between the MIES and a general purpose optimizer is also given.

10.5.1 Overview

The objective of the optimization problem is to find the materials (represented by categorical variables) and lamination angles (continuous variables) in order to minimize the bending of a loaded plate composed of five layers.

The available materials and their properties are reported in Table 10.1. As one can see, the list includes orthotropic and isotropic materials. In case of orthotropic materials, the stiffness of the material is crucially affected by the lamination angle. Contrarily, isotropic materials have equal in-plane and out-of-plane Young's modulus. Hence, the lamination angle of isotropic materials does not affect the material behaviour. The use of both types of materials considerably increases the difficulty of the problem from the modeling perspective: The importance of the continuous parameters depends on the value of the categorical variables.

The plate has been loaded by a lifting load, which is linearly distributed along the length of the plate applied on the nodes in the centreline. An encastre at the root of the plate has been enforced as shown in Fig. 10.2. With the intent to reproduce

Table 10.1 Properties of the material used in this study

Properties		Young's Modulus 0	Young's Modulus 90	In-plane Shear modulus	Poisson's ratio	
Symbols		E1	E2	G12	v12	Density
Units		GPa	GPa	GPa		g/cc
CF	Vf 50%	70	70	5	0.10	1.60
HMCF	Vf 50%	85	85	5	0.10	1.60
E glass	Vf 50%	25	25	4	0.20	1.90
Kevlar	Vf 50%	30	30	5	0.20	1.40
Std CF	Vf 60%	135	10	5	0.30	1.60
HMCF	Vf 60%	175	8	5	0.30	1.60
M55**	Vf 60%	300	12	5	0.30	1.65
E glass	Vf 60%	40	8	4	0.25	1.90
Kevlar	Vf 60%	75	6	2	0.34	1.40
Boron	Vf 60%	200	15	5	0.23	2.00
Steel	S97	207	207	80	0.3	7.85
AL	L65	72	72	25	0.3	2.7

** Calculated figures

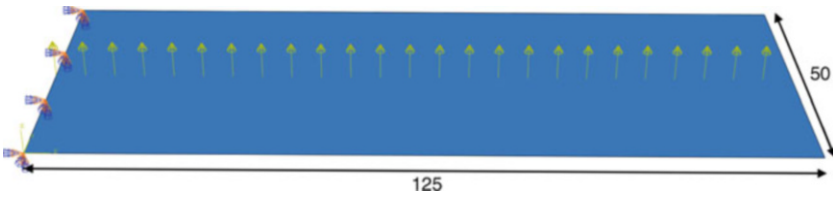


Fig. 10.2 Load and boundary constrains applied to the multi-laminate plate

uncertainty due to manufacturing tolerances and measurement conditions, perturbations to the nominal values have been added to both lamination angles and load magnitudes.

The purpose of this test case is to point out the difficulties of handling an “expensive” (≈ 10 s for each run), multi-modal, mixed-integer, “high-dimensional” (5 continuous + 5 categorical variables) problem under uncertainty with an extremely limited budget (150 evaluations).

10.5.2 Optimization Problem

The objective of the problem is the minimization of the displacement along the loaded axis of one of the vertices of the multi-layered plate tip s_t (red point in the corner of the plate in Fig. 10.3). The optimization problem is defined as

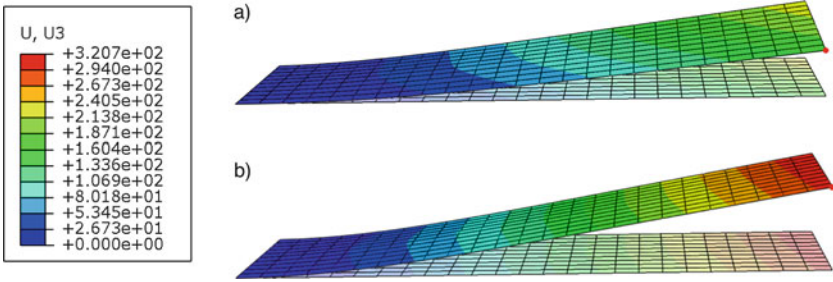


Fig. 10.3 Contour of the displacement in the loaded direction in the best configurations obtained employing DE (a) and MIES (b)

$$\begin{aligned} & \min_{x \in \mathbb{R}^5, y \in \mathbb{D}^5} f(x, y) = s_t \\ & \text{subject to } \begin{cases} r_i \in [-89, 90], 1 \leq i \leq 5 \\ \mathbb{D}^5 = D_1 \times \dots \times D_5 \\ d_i = \{d_i^1, \dots, d_i^{12}\}, 1 \leq i \leq 12 \end{cases} \end{aligned}$$

where d_i^1, \dots, d_i^{12} are the 12 available materials [39]. The first five variables correspond to the lamination angles, and the latter five describe the material of each ply as categorical variables. The categorical variables are mapped to integers from 1 to 12 to allow a numerical optimizer based on differential evolution (DE) [2] to handle them.

Since the mass of each ply has been fixed, the thickness will be dependent on the material density.

10.5.3 Methodology

To perform the optimization process, we take advantage of surrogate modeling techniques and rely on the SPOT R package. The experiments have been conducted employing MIES and DE as optimizers (on the surrogate model). As shown in the previous sections, MIES is dedicated to handling mixed variables, including categorical variables. Contrarily, DE is not designed to handle discrete or categorical variables. In order to compute an accurate estimation of the plate maximal displacement, the finite element analysis solver Abaqus [23] has been employed. Therefore, the optimization problem requires a coupling between SPOT and Abaqus.

The process follows the algorithm described in Algorithm 1: Initially a design is created by LHS. All the candidates in the design are then evaluated with the objective function (via Abaqus). This function receives the candidate solutions that describe the characteristics of the plate and feeds the corresponding displacement

back to the optimizer. Nevertheless, to simulate the uncertainty, the nominal lamination angles and loads magnitudes are perturbed randomly by Gaussian perturbations with mean 0 and variance, respectively, 1 and 0.1. The candidates and the function responses are used to train a Kriging model. Then, an optimizer searches for the most promising candidate solution by optimizing an infill criterion based on the model. Here, the criterion is the expected improvement. Based on the assumption of expensive function evaluations, a very small budget of 150 function evaluations has been used.

10.5.4 Results

The obtained results clearly highlight that mixed-discrete problems present peculiar challenges. Such issues are unlikely to be resolved by continuous optimizers. A convergence plot is shown in Fig. 10.5. It can be seen that the best configuration found by MIES outperforms the one found by DE. Although the processes have been started from the same design, the results differ by around 80% (Table 10.2). However, to have a statistically meaningful comparison, we should repeat the analyses with the two best designs in order to quantify the effect of the uncertainties.

The displacements of the two best plate configurations are shown in Fig. 10.3 with the same scale and magnification factor. Figure 10.4 shows that the configurations differ significantly, concerning both the continuous and categorical variables. Particularly, it is worth to focus on the effect of the lamination angles: The configuration found using DE presents materials that are mostly aligned with the plate. This configuration would lead to the maximal uniaxial stiffness. In this case, the displacement would be equal for all the nodes lying on the tip. However, the

Table 10.2 Optimal design achieved using DE and MIES

Optimal design	Displacement	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
DE	169.49	0.35	15.04	-23.496	5.18	75.43	9	4	9	7	6
MIES	303.34	-5.79	-89.00	-89.00	-35.27	-7.15	7	1	7	9	7

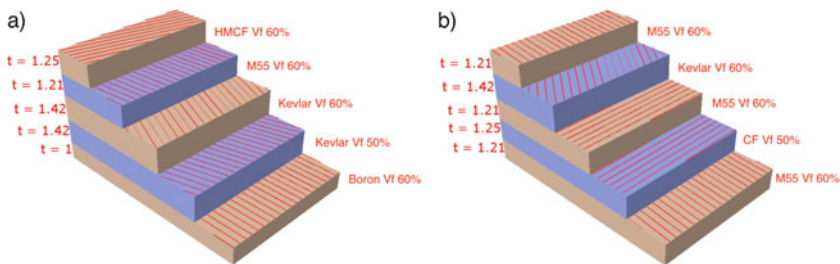


Fig. 10.4 Best lamination configuration (lamination angles, materials and thickness) obtained employing DE (a) and MIES (b)

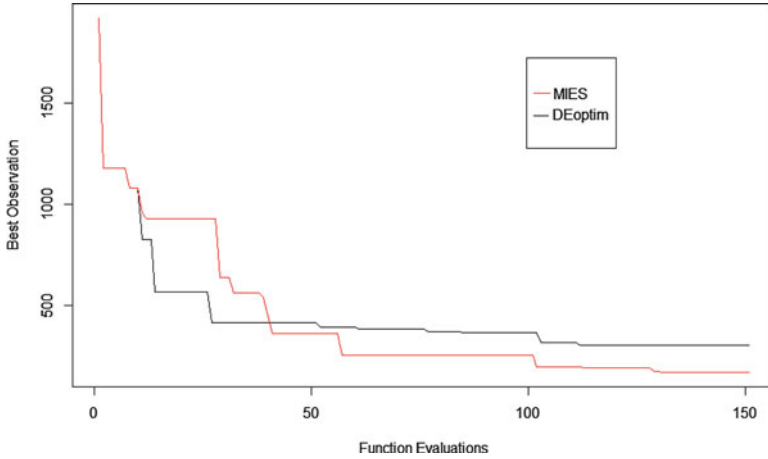


Fig. 10.5 Evolution of the best observation during the optimization processes

object of the optimization is the displacement in the corner of the plate, which is at some distance from the centreline of the plate. This means that another minimum is present, besides the local minimum resulting from the maximization of the stiffness in the direction of the plate. The second minimum is the global one. It consists in the perfect balance between the stiffness in both the directions in the plane of the plate. The balance is determined on the basis of the ratio between the length and the height of the plate. As a result, the overall displacement at the tip will be higher, but the displacement at the corner will be lower (Fig. 10.5).

In fact, in the configuration resulting from MIES, two plies are laminated with angles that are orthogonal to the direction of the plate. In light of this observation, we can say that this problem is clearly multi-modal in regard to the continuous variables. The multi-modality of the function derives from the dependence of both the mechanical properties and the ply thickness on the chosen material. This is also reflected in Fig. 10.6, where the interactions of four important variables are depicted. In each plot, two variables are varied, while the remaining two are fixed to their optimal values. The red dots represent the observed values. The dataset used to train this model is composed of the observations made during the optimization process using MIES. In Fig. 10.6a and b, the interaction between the lamination angles and the materials for both the plies are shown. In both the cases, two distinct regions of well-performing configurations are present. In Fig. 10.6c and d, the interaction between the materials and their lamination angles of the two plies are represented. The figures clearly show a complex and multi-modal search landscape. In light of these considerations, one can see the complexity involved in this, apparently, simple problem.

In the last section of this chapter, an example of the application of SPOT on a real-world optimization problem has been reported and analysed. The complexity of mixed-discrete real-world optimization problems has been addressed in this chapter. Despite the difficulties, more efficient ways to handle them have been developed.

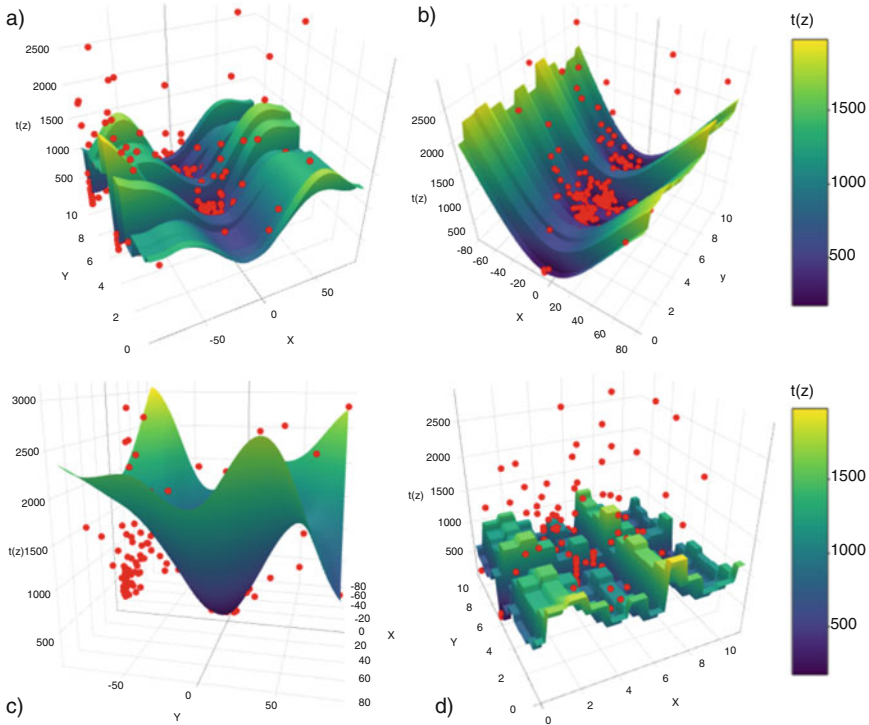


Fig. 10.6 Visualizations of the objective function landscapes in respect of variable combinations concerning Ply 1 and Ply 5. For each individual plots, variables that are not shown are fixed to the respective optimal values. **(a)** Material and lamination angle of Ply 1. **(b)** Material and lamination angle of Ply 5. **(c)** Lamination angles of Ply 1 and Ply 5. **(d)** Materials of Ply 1 and Ply 5

Nevertheless, despite the cutting-edge algorithms, these problems still appear very complex to tackle.

Acknowledgments This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

References

1. C.C. Aggarwal, A. Hinneburg, D.A. Keim, On the surprising behavior of distance metrics in high dimensional space, in *Database Theory ICDT 2001* (Springer Science + Business Media, Berlin, 2001), pp. 420–434
2. D. Ardia, K. Boudt, P. Carl, K.M. Mullen, B.G. Peterson, Differential Evolution with DEoptim: an application to non-convex portfolio optimization. *R J.* **3**(1), 27–34 (2011)

3. M. Asadi, J. Goldak, Combinatorial optimization of weld sequence by using a surrogate model to mitigate a weld distortion. *Int. J. Mech. Mater. Des.* **7**(2), 123–139 (2011)
4. L. Bajer, M. Holeňa, Surrogate model for continuous and discrete genetic optimization based on RBF networks, in *Intelligent Data Engineering and Automated Learning – IDEAL 2010*. Lecture Notes in Computer Science, vol. 6283 (2010), pp. 251–258
5. J.W. Bandler, R.M. Biernacki, S.H. Chen, P.A. Grobelny, R.H. Hemmers, Space mapping technique for electromagnetic optimization. *IEEE Trans. Microw. Theory Tech.* **42**(12), 2536–2544 (1994)
6. T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation—The New Experimentalism*. Natural Computing Series (Springer, Berlin, Heidelberg, New York, 2006)
7. T. Bartz-Beielstein, S. Markon, Tuning search algorithms for real-world applications: a regression tree based approach, in *Proceedings 2004 Congress on Evolutionary Computation (CEC'04)*, Portland, OR, ed. by G.W. Greenwood (IEEE, Piscataway NJ, 2004), pp. 1111–1118
8. T. Bartz-Beielstein, M. Zaefferer, Model-based methods for continuous and discrete global optimization. *Appl. Soft Comput.* **55**, 154–167 (2017)
9. T. Bartz-Beielstein, L. Gentile, M. Zaefferer, In a nutshell: sequential parameter optimization. Technical report, TH Köln, 2017
10. A. Bergman, M.W. Feldman, Recombination dynamics and the fitness landscape. *Phys. D: Nonlinear Phenom.* **56**(1), 57–67 (1992)
11. B. Borchers, J.E. Mitchell, An improved branch and bound algorithm for mixed integer nonlinear programs. *Comput. Oper. Res.* **21**(4), 359–367 (1994)
12. L. Breiman, Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
13. C. Ding, X. He, H. Zha, H.D. Simon, Adaptive dimension reduction for clustering high dimensional data, in *2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings* (IEEE, New York, 2002), pp. 147–154
14. N.R. Draper, H. Smith, *Applied Regression Analysis*, vol. 326 (Wiley, New York, 2014)
15. M.A. Duran, I.E. Grossmann, An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Math. Program.* **36**(3), 307–339 (1986)
16. R. Filomeno Coelho, M. Herrera, M. Xiao, W. Zhang, On-line metamodel-assisted optimization with mixed variables, in *Evolutionary Algorithms and Metaheuristics in Civil Engineering and Construction Management*. Computational Methods in Applied Sciences, vol. 39, ed. by J. Magalhães-Mendes, D. Greiner (Springer International Publishing, Basel, 2015), pp. 1–15
17. C.A. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications* (Oxford University Press, Oxford, 1995)
18. L. Fonseca, H. Barbosa, A. Lemonge, A similarity-based surrogate model for expensive evolutionary optimization with fixed budget of simulations, in *Proceedings of the Congress on Evolutionary Computation (CEC'09)* (IEEE, New York, NY, 2009), pp. 867–874
19. A. Forrester, A. Keane et al., *Engineering Design via Surrogate Modelling: A Practical Guide* (Wiley, Chichester, 2008)
20. A.M. Geoffrion, Generalized benders decomposition. *J. Optim. Theory Appl.* **10**(4), 237–260 (1972)
21. T. Hemker, Derivative Free Surrogate Optimization for Mixed-Integer Nonlinear Black Box Problems in Engineering. PhD thesis, Technische Universität Darmstadt, December 2008
22. M. Herrera, A. Guglielmetti, M. Xiao, R. Filomeno Coelho, Metamodel-assisted optimization based on multiple kernel regression for mixed variables. *Struct. Multidiscipl. Optim.* **49**(6), 979–991 (2014)
23. K. Hibbitt, *Abaqus: User's Manual: Version 6.13: Hibbitt* (Karlsson & Sorensen, Incorporated, Providence, RI, 2013)
24. R.R. Hocking, A biometrics invited paper. the analysis and selection of variables in linear regression. *Biometrics* **32**(1), 1–49 (1976)
25. F. Hutter, H.H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration (extended version). Technical Report TR-2010-10, University of British Columbia, Department of Computer Science, 2010. Available online <http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf>

26. Y. Jin, J. Branke, Evolutionary optimization in uncertain environments—a survey. *IEEE Trans. Evol. Comput.* **9**(3), 303–317 (2005)
27. R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct. Multidiscipl. Optim.* **23**(1), 1–13 (2001)
28. D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive Black-Box functions. *J. Global Optim.* **13**, 455–492 (1998)
29. T. Kohonen, The self-organizing map. *Proc. IEEE* **78**(9), 1464–1480 (1990)
30. R. Li, M.T.M. Emmerich, J. Eggermont, E.G.P. Bovenkamp, T. Bäck, J. Dijkstra, J.H.C. Reiber, Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis, in *2008 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, New York, 2008), pp. 2764–2771
31. R. Li, M.T. Emmerich, J. Eggermont, T. Bäck, M. Schütz, J. Dijkstra, J.H. Reiber, Mixed integer evolution strategies for parameter optimization. *Evol. Comput.* **21**(1), 29–64 (2013)
32. D. Liu, A. Litvinenko, C. Schillings, V. Schulz, Quantification of airfoil geometry-induced aerodynamic uncertainties—comparison of approaches. *SIAM/ASA J. Uncertain. Quant.* **5**(1), 334–352 (2017)
33. R. Lohmann, Structure evolution and incomplete induction. *Biol. Cybern.* **69**(4), 319–326 (1993)
34. M.D. McKay, R.J. Beckman, W.J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
35. J. Močkus, On Bayesian methods for seeking the extremum, in *Optimization Techniques IFIP Technical Conference* (1974), pp. 400–404
36. D.C. Montgomery, *Design and Analysis of Experiments* (Wiley, New York, 2017)
37. A. Moraglio, A. Kattan, Geometric generalisation of surrogate model based optimisation to combinatorial spaces, in *Proceedings of the 11th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP'11* (Springer, Berlin, Heidelberg, 2011), pp. 142–154
38. M.D. Morris, Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**(2), 161–174 (1991)
39. P. Composites Ltd. Mechanical properties of carbon fibre composite materials. http://www.performance-composites.com/carbonfibre/mechanicalproperties_2.asp. Date: 2018-07-01
40. G. Papa, Parameter-less algorithm for evolutionary-based optimization. *Comput. Optim. Appl.* **56**(1), 209–229 (2013)
41. P.A. Romero, A. Krause, F.H. Arnold, Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci.* **110**(3), E193–E201 (2013)
42. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985
43. A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, *Sensitivity Analysis in Practice* (Wiley, New York, 2004)
44. M. Schütz, J. Sprave, Application of parallel mixed-integer evolution strategies with mutation rate pooling, in *Proceedings of the Fifth Annual Conference on Evolutionary Programming* (1996). Citeseer
45. H.-P. Schwefel, Evolution and optimum seeking. sixth-generation computer technology series (1995)
46. S. Shan, G.G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Struct. Multidiscipl. Optim.* **41**(2), 219–241 (2010)
47. J. Smith, C. Stone, M. Serpell, Exploiting diverse distance metrics for surrogate-based optimisation of ordering problems: a case study, in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, GECCO '16*, pp. 701–708 (ACM, New York, NY, 2016)

48. J. Sobieszczanski-Sobieski, Sensitivity analysis and multidisciplinary optimization for aircraft design—recent advances and results. *J. Aircraft* **27**(12), 993–1001 (1990)
49. I. Sobol, Y.L. Levitan, A pseudo-random number generator for personal computers. *Comput. Math. Appl.* **37**(4–5), 33–40 (1999)
50. C. Teixeira, J. Covas, T. Stützle, A. Gaspar-Cunha, Optimization of co-rotating twin-screw extruders using pareto local search, in *Advances in Intelligent and Soft Computing* (Springer Science + Business Media, Berlin, 2010), pp. 3–10
51. C. Teixeira, J.A. Covas, T. Stützle, A. Gaspar-Cunha, Multi-objective ant colony optimization for the twin-screw configuration problem. *Eng. Optim.* **44**(3), 351–371 (2012)
52. J. Tu, D. Jones, Variable screening in metamodel design by cross-validated moving least squares method, in *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2003), pp. 1669
53. U. Utecht, K. Trint, Mutation operators for structure evolution of neural networks, in *International Conference on Parallel Problem Solving from Nature* (Springer, Berlin, 1994), pp. 492–501
54. I. Voutchkov, A. Keane, A. Bhaskar, T.M. Olsen, Weld sequence optimization: The use of surrogate models for solving sequential combinatorial problems. *Comput. Methods Appl. Mech. Eng.* **194**(30–33), 3535–3551 (2005)
55. T. Weise, M. Zapf, R. Chiong, A.J. Nebro, Why is optimization difficult? in *Nature-Inspired Algorithms for Optimisation* (Springer, Berlin, 2009), pp. 1–50
56. T. Westerlund, F. Pettersson, An extended cutting plane method for solving convex minlp problems. *Comput. Chem. Eng.* **19**, 131–136 (1995)
57. D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
58. M. Zaefferer, J. Stork, T. Bartz-Beielstein, Distance measures for permutations in combinatorial efficient global optimization, in *Parallel Problem Solving from Nature—PPSN XIII*, ed. by T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (Springer, Cham, 2014), pp. 373–383
59. M. Zaefferer, J. Stork, M. Friese, A. Fischbach, B. Naujoks, T. Bartz-Beielstein, Efficient global optimization for combinatorial problems, in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO '14* (ACM, New York, NY, 2014), pp. 871–878

Chapter 11

Parameter Control in Evolutionary Optimisation



Margarita Antoniou, Rok Hribar, and Gregor Papa

Abstract Finding the global optimum of a complex function is one of the long-standing goals of applied mathematics and numerical analysis. Evolutionary algorithms have become a popular way of solving demanding and expensive optimisation problems. These algorithms are composed of several control parameters that need to be set, for the procedure of searching for the optimum of an objective function to be successful. Parameter setting is a challenging topic, since control parameter values affect significantly the performance of the algorithm. Moreover, the control parameters may interact with each other in an unpredictable way. On top of this, at different stages of optimisation process, different control parameters may be needed. In this chapter, we introduce some basic control parameters that can be modified and the main methods of parameter settings, i.e. parameter tuning (offline) and parameter control (online). More focus is given on parameter control and the three different strategies used to implement it: the deterministic, the adaptive and the self-adaptive parameter control. In addition, a short comparison between parameter tuning and parameter control is given, based on the current literature. The last section refers to the improvement that parameter control can bring in some of the most complex instances of real-world optimisation problems, such as dynamic or problems under uncertainty, when using EAs to solve them and some of the strategies that can be used in each instance.

Keywords Parameter control · Parameter tuning · Evolutionary optimisation · Adaptation

M. Antoniou (✉) · R. Hribar · G. Papa
Jožef Stefan Institute, Ljubljana, Slovenia

Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
e-mail: margarita.antoniou@ijs.si; rok.hribar@ijs.si; gregor.papa@ijs.si

11.1 Evolutionary Optimisation

Finding the exact solution of an optimisation problem with a deterministic method guarantees the optimality of this solution. For example, running an exhaustive search of all the combinations of the variables of a function, until the best solution is obtained, is one of these approaches. As one can understand, this is prohibitive to problem instances above a number of variables, since the runtime complexity grows exponentially, the search space becomes huge and the needed computational resources are not yet available. Moreover, one should consider that usually what is being optimised (especially in real-world applications) is actually a model of the problem, a representation of reality, which already is an approximation. Therefore, trying to find an exact solution is not always useful, since finding a solution near to optimal or a better than known so far can be sufficient enough. Stochastic algorithms that use random sampling for directing the search process [1] cannot prove the convergence to a global optimum, but can provide a sufficiently good solution and alternative for this kind of complex problems within a reasonable time [2]. Moreover, metaheuristic algorithms make few or no assumptions about a problem and may apply a deterministic method to explore the search space. A metaheuristic algorithm may implement single solution or population-based searches. In a single solution approach, one single candidate solution is being improved, whilst in population-based methods a number of candidate solutions are taken into consideration to guide the search [3]. One of the most popular branches of population-based metaheuristics is evolutionary algorithm (EA).

11.1.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) are nature-inspired approaches, imitating the Darwinian evolution of species or other similar phenomena. They occupy an important position amongst optimisation techniques when solving difficult optimisation problems, being non-linear, non-convex, multimodal or non-differentiable in the variable search space, or might even require considering multiple contradictory objectives. They are typically used to provide good solutions to problems that cannot be solved easily using other techniques. When it may be too computationally intensive to find an exact solution, a near-optimal solution might be sufficient. EAs do not guarantee to find an optimal solution, but they often find a good and acceptable solution if it exists [4].

EAs apply principles of evolution found in nature, such as reproduction, mutation, recombination and selection. In an EA a number of artificial individuals (candidate solutions to the optimisation problem) search simultaneously over the problem search space [5]. The shared environment dictates the fitness or performance of each individual in the population. The individuals compete continually with each other to discover optimal areas of the search space. It is expected that

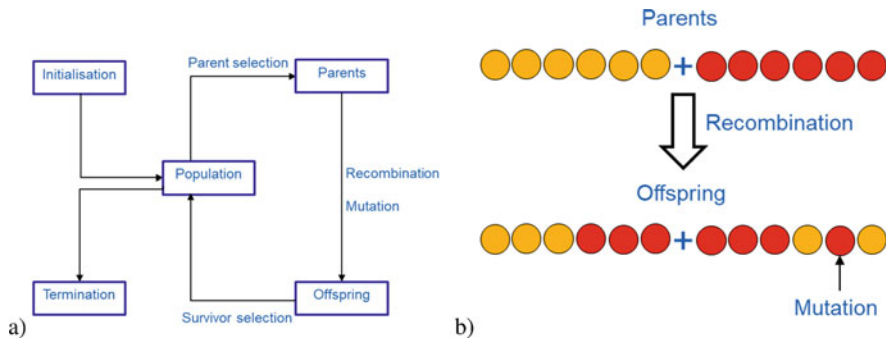


Fig. 11.1 Simplified flowchart of an evolutionary algorithm: (a) Parents are selected from the population and produce offspring, through mutation and recombination. The most suitable are selected to form the next population (inspired by [4]). (b) The chromosomes of two parents, the yellow and red dots, are recombined and mutated to produce two unique offspring

over time the most successful of these individuals will evolve to discover the optimal solution. Each iteration of an EA involves a competitive selection that filters out weak solutions. The fittest individuals are more likely to be chosen for reproduction and being modified by crossover and mutation, eventually leading potentially to superior ones. The selected solutions are recombined with other solutions by swapping parts of a solution with another (crossover). The solutions can further be mutated by making a small change to a single element of the solution (mutation).

A simplified flowchart of one of the most popular EA, i.e. genetic algorithm (GA), is shown in Fig. 11.1, along with the concept of chromosome representation of a candidate solution, crossover, and mutation in an illustrative way.

11.1.2 Exploration and Exploitation

Exploration and exploitation are the two main aspects of evolutionary problem-solving [6]. The exploration and exploitation of a search space must be addressed by every search algorithm. Exploration is the process of visiting completely new regions of a search space, while exploitation is the process of visiting those regions of a search space within the neighbourhood of previously visited points (see Fig. 11.2). A search algorithm needs to establish a good ratio between exploration and exploitation in order to be successful.

Different metaheuristics present unique exploration and exploitation capabilities, i.e. they have various forms of exploring and exploiting the search space. What works to solve a particular problem may be not good for tackling another one. Moreover, each problem can demand a particular set of control parameters for each algorithm. In this context, adaptive algorithms have appeared trying to solve as

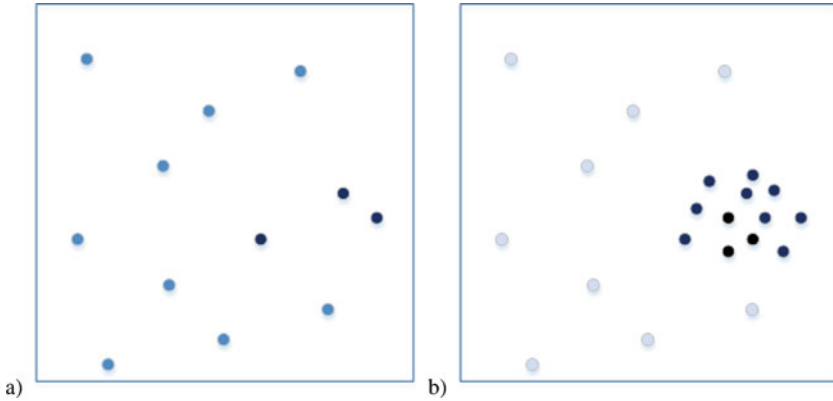


Fig. 11.2 Difference between exploration and exploitation: (a) During the exploration the entire search space is checked and some promising regions are detected (i.e. darker dots that present better solutions). (b) During the exploitation the search focus is on the promising region and the nearby of the promising (better) solutions is further investigated to find even better solutions

many problems as possible with no algorithm code changes. Today, approaches deal with adaptation of operators [7] or parameters [8]. When dealing with operators, the adaptation tries to identify which operator (or its implementation) is more suitable to the problem, whilst in parameters, the algorithm attempts to discover the best control parameter value. Both adaptation situations are usually implemented separately (not concurrently in the same algorithm), but they are both executed on-the-fly, during the optimisation process. This chapter covers only the latter case of the control parameter setting.

11.1.3 The Role of Control Parameters

The EAs are driven by control parameters (examples in Sect. 11.2), which are crucial for their efficient performance [9]. The best control parameter values depend on the problem and by the smart encoding of the variables the level of the problem difficulty might change. In addition, the EAs should be robust, i.e. the control parameter configuration should allow stable behaviour regardless of the problem instance.

The control parameters can be either tuned offline (before the actual optimisation process) or adapted online (during the optimisation process); see details in Sect. 11.3. Whilst the fine-tuned algorithm's control parameters allow robust behaviour (especially for static problems), dynamic modifications are required on control parameters to more effectively exploit and explore the search space [10]. The latter case includes adaptive mechanisms on control parameters, since the best

algorithm control parameter values depend on the current state of the optimisation process and thus change over time [11, 12].

As the adaptation of the algorithm control parameters depends on different scenarios, one should consider their influence (further details are available in Sect. 11.4): deterministic (time-dependent, feedback-free), self-adaptive (encoding control parameters with other variables) and adaptive (using statistical feedback from the optimisation process). The automatic setup of algorithms and their control parameters is one of the prerequisites to allow ease of use of the complex industrial optimisation tools.

One should differentiate between *control parameter* (Sect. 11.2), which denotes the parameter of the algorithm that drives the algorithm's behaviour, and the *parameter control* (Sect. 11.3), which denotes the online process of changing the control parameter(s). The main goals of parameter control are the identification of good control parameter values on-the-fly and tracking of good control parameter values as they change during the optimisation process [13].

11.2 Control Parameters

This section presents typical control parameters that can be found in evolutionary algorithms, along with their characteristics. This is followed by a description of their influence on the algorithm's performance and also the interdependence amongst different control parameters is presented.

11.2.1 Typical Control Parameters

There are obviously different control parameters [14], with different effects, for different optimisation algorithms. As an example, for the most typical implementations of EAs, one can adapt:

- Population size
 - Also referred as μ .
 - The number of individuals (solutions) that are considered (and evaluated) in parallel and are combined to form new solutions.
 - In general, larger population size gives larger variability within the population, but increases the computational time and slows down the convergence (when the number of generations remains the same).
- Offspring size
 - Also referred as λ .
 - The number of offspring (resulting solutions) that are produced at each generation.

- In general, larger offspring size speeds up the search process and increases the diversity of solutions (in earlier stages of the optimisation process).
- Mutation rate or mutation probability
 - Also referred as p_m .
 - The frequency of new mutations in a single gene (variable representation) or individual (solution representation) over time.
 - In general, higher mutation rate increases the versatility of the solutions within the population, but it also increases the amount of randomness in the search.
- Mutation step size
 - Also referred as σ .
 - The size of the change of mutation rate compared to its previous value. In the case of real-valued search spaces, mutation is usually performed by adding a normally distributed random value to each optimised variable.
 - In general, higher mutation step size has a similar influence as the higher mutation rate.
- Crossover rate or crossover probability
 - Also referred as p_c .
 - The probability that crossover is performed between two chosen solutions.
 - In general, lower crossover probability prevents exchange of genetic material and causes slower convergence.
- Selection pressure
 - It is the degree to which the better individuals are favoured. Which solutions will provide the individual with an increased chance of surviving over others, i.e. to be forced to contribute to the next generation. Therefore, individuals with certain phenotypes have an advantage when it comes to survival and reproduction.
 - The higher the selection pressure, the more good solutions are kept from generation to generation. It is related to elitism strategy, which ensures good solutions to proceed to the next generation.
- Tournament size
 - Tournament selection involves running tournaments amongst pairs (or groups) of individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for the next phase (i.e. crossover).
 - When tournament size is higher, the weaker solutions have a smaller chance to be selected.
- Number of generations
 - Also referred as n_g .

- The number of iterations for reproduction and recombination of solutions during the optimisation process.
- It is very often connected to the population size, especially when the total number of evaluations is limited.
- In general, when using a higher number of generations, it is more probable that the optimisation process will converge towards a better/optimal solution, but also the computational time is increased.

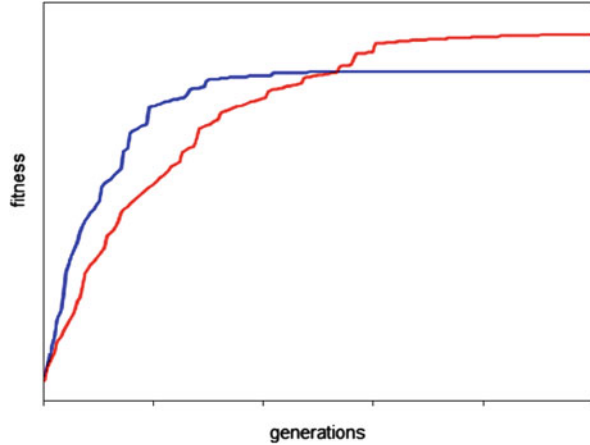
Whenever some control parameter or other component of the optimisation algorithm is changed, then, depending on the implementation, it can influence either an individual solution, or it can influence the whole population of solutions.

11.2.2 What Else can be Adapted

Further on, to respond to changes and to improve the optimisation convergence, one can also adapt:

- Fitness function
 - Sometimes it is needed either to adapt to changing environment with modified fitness function, or it is needed to add some constraints into the fitness function (e.g. penalty terms for constraints) [15].
- The representation of the problem
 - The representation of the solution can be changed to reflect either the simplifications due to already optimised parts of the solution or to increase the force of promising parts for the next generations.
 - The proper encoding (representation of the problem) ensures that the optimised variables are modified efficiently and that the heuristics can be easily applied during the optimisation. Different representations of the same problem can give a different insight into the properties, including redundant encodings, synonymity, locality and connectivity, as well as their interrelationships [16].
- The components of the algorithm
 - An evolutionary algorithm can use different combinations of parent selection, crossover operators, mutation operators, survival selection and termination condition [17]. For example, the implemented versions of crossover and mutation operators influence the way how the encoded information is modified.

Fig. 11.3 Influence of different algorithm setting on its performance. The same components (selection, crossover, mutation) and random seeds are used, but different values of control parameters result in faster convergence (blue line), or an algorithm may reach higher fitness value (red line)



11.2.3 Influence on Algorithm Performance

The performance of an optimisation algorithm can be measured in terms of execution time, robustness of solutions, resource utilisation, convergence, etc. Besides the control parameters the performance of an evolutionary algorithm also depends on interactions with other issues (representation, components).

An exemplary Fig. 11.3 presents the general comparison of fitness convergence for two different control parameter settings of the algorithm. Even if the same components (selection, crossover, mutation) are used and the same random seed is used for EA, different values of control parameters may result in faster initial convergence (blue line), or an algorithm may reach a higher fitness value (red line).

11.2.4 Interaction of Control Parameters

Based on [18], Fig. 11.4 presents the fitness convergence for several different settings of the control parameters (see Table 11.1 for details). The achieved fitness is different depending on different probabilities of crossover and mutation, whilst number of generations and population size are fixed. For a given problem of optimising the design of integrated circuit (i.e. elliptic filter consisting of 34 operations), it is shown that the optimal tuned control parameters might be $\mu = 10$, $n_g = 100$, $p_c = 0.7$ and $p_m = 0.01$. Other combinations of control parameter values might not give such good results. Therefore, optimal setting of the optimisation algorithm is preferred; however, finding these optimal values is a difficult, but important, task.

Fig. 11.4 Interdependence between probabilities of crossover and mutation (with fixed number of generations and population size). For optimising the design of an elliptic filter [18], it is shown that the optimal tuned (ensuring highest quality) control parameters might be $\mu = 10$, $n_g = 100$, $p_c = 0.7$ and $p_m = 0.01$

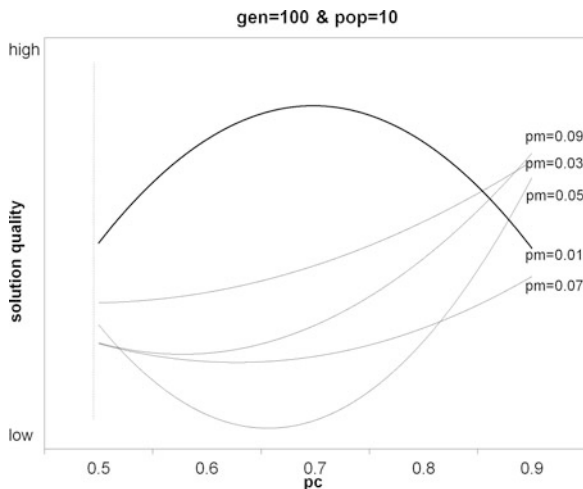


Table 11.1 Control parameter settings as used in [18]

Control parameter	Values
Population size	$\mu = 10$
Number of generations	$n_g = 100$
Probability of crossover	$p_c = 0.5, 0.6, 0.7, 0.8, 0.9$
Probability of mutation	$p_m = 0.01, 0.03, 0.05, 0.07, 0.09$

11.3 Setting Approaches

The setting of the control parameters can be classified into two major approaches [19], as presented in Fig. 11.5. The first one is parameter tuning, where the parameters are tuned in advance of the actual optimisation process. The second one is parameter control, where the parameters are modified online, during the optimisation process. In the latter case there are different strategies for the adaptation of the control parameters.

Note that some newer works [13] suggest a slightly modified classification for parameter control. Instead of making a distinction only amongst deterministic, adaptive and self-adaptive strategy, they suggest a classification into state-dependent, success dependent, learning-dependent, self-adaptive and hyper-heuristics strategies. As currently most of the literature still relies on the initial classification of [19], this chapter follows that one, too.

For **parameter tuning** (see Sect. 11.3.1 for details) a large amount of literature on finding good **static control parameter** values exists [20–23]. There are even some frameworks to allow easy parameter tuning, a reference list of which is being mentioned in the section below. The so-called parameter tuning mechanisms involve checking several combinations and evaluating their efficiency. However, optimal static parameter values for one problem may be much different for other similarly looking problems. Namely, a small change in one parameter can cause

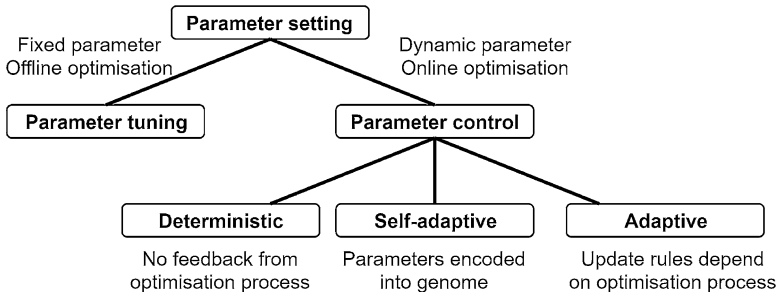


Fig. 11.5 Classification of parameter setting [19]

huge performance gaps. For example, changing the mutation rate by a small constant factor might drastically (exponentially) change the running time.

Within **parameter control** (see Sects. 11.3.2 and 11.4 for details) the best control parameter values depend on the current state of the optimisation process and thus change over time (**non-static control parameter**). There are several simple methods for allowing algorithms to set good control parameter values by themselves, e.g. (1+1) EA_α [24], PLES [12], CMA-ES [25] and jDE [26] (more details in Sect. 11.4).

11.3.1 Parameter Tuning

A lot of research exists on analysing ways to tune parameters in a smart, efficient and effective way. Mostly they suggest/recommend some values, which have turned to be robust enough for the tested problems. The recommendations for population size, mutation and crossover probabilities, selection strategies, etc. are either:

- **Absolute values**, which are independent of problem class or problem size [27] [28]. For example, for a problem with five variables, the population size should be 10, and the number of generations should be 100, whilst crossover and mutation probabilities should be 0.8 and 0.02, respectively.
- **Relative values**, which depend on some property of the problem [29], e.g. $1/n$ as mutation rate for problems of lengths n (i.e. chromosome with encoded n variables), or the ratio $\mu/\lambda \approx 1/4$ [30].

This approach works well for a broad range of problems, but unfortunately problem size is not the only feature that influences the efficiency of the search. Some more modern tuning approaches run a number of initial tests and observe the performance of different control parameter values, and finally they choose the control parameter values that seem to be the most promising. The list of such parameter tuning tools include:

- *irace*—a software package with a number of automatic configuration procedures. Its iterated procedures have been used to automatically configure various algorithms [31].
- *SPOT*—the sequential parameter optimisation package for R is a toolbox for tuning and understanding simulation and optimisation algorithms [32]; see also Chap. 10 for details.
- *GGA*—the gender-based genetic algorithm configurator, integrated with surrogate model for predicting high-performance regions in the parameter space [33].
- *ParamILS*—an automatic framework for the identification of performance-optimising parameter settings [34].
- *SMAC*—the framework constructs explicit regression models to describe the dependence of target algorithm performance on control parameter settings [35].
- *Spearmint*—software package to perform Bayesian optimisation which can be used to find appropriate control parameter values in as few runs as possible [36].

The automated identification of control parameter values that these tools possess recommends efficient values for static problems, where unchanged values of control parameters are used through the whole optimisation process.

11.3.2 *Parameter Control*

The efficiency of optimisation with specific control parameter values depends on the characteristics of the search space landscape. Some parameter values might be good for separable functions, others for multimodal, convex, discontinuous and so on. Usually, when population moves through the search space, such characteristics change. For example, at the beginning of optimisation, the fitness landscape might appear highly multimodal and irregular, but then the population converges towards a region that contains a very narrow valley. In order for the algorithm to follow this narrow valley efficiently, it needs to employ a different strategy than when the landscape appeared multimodal. In this case different control parameter values are appropriate for different stages of the optimisation process. This indicates that no globally optimal control parameters exist and that one should use non-static parameters that adapt during the search/optimisation process.

Moreover, different stages of the optimisation process require different control parameter values [37]. It can be easily shown that exploration and exploitation phases of the search process require different values of control parameters:

- In the beginning, in the exploration phase, large mutation rates and small selective pressure are needed to make large jumps across the search space and discover different areas of the search space.
- Towards the end, in the exploitation phase, small mutation rates and high selective pressure are required to focus the search around the promising discovered regions of the search space.

The right choice of control parameter values depends on the problem and even the problem instance that is solved and the algorithm that is employed for the search. So, whenever a new optimisation problem or its new/different instance is introduced or a new optimisation algorithm is employed, a new set of control parameter values is needed. Surely, some preliminary experiments are often conducted to find (i.e. tune) reasonable initial control parameter values. However, based on the implementation of the adaptation, such tuning might not even be necessary since some types of parameter control can find appropriate control parameter values in the early stage of optimisation by themselves. The next section presents more details on different adaptation strategies.

11.4 Parameter Control Strategies

When deciding on the proper parameter control strategy, several decisions have to be made. First, one needs to decide what is the update trigger, which can be one of the following:

- Number of fitness evaluations performed
- Time elapsed
- Progress in terms of absolute or relative fitness values
- Diversity measures

Next, it should be decided which strategy should be used to update the control parameter values (their details are presented in the following sections):

- **Deterministic** (or time-dependent)
 - It is usually time-dependent and the algorithm does not consider any feedback from the optimisation process.
- **Adaptive**
 - Here some statistical feedback from the optimisation process is used to determine the direction and magnitude of change of the control parameter value.
- **Self-adaptive**
 - It treats control parameters as part of the optimisation (being a complex optimisation problem) and EAs are used to find good values of the control parameters (i.e. control parameters are encoded together with other variables and are concurrently optimised).

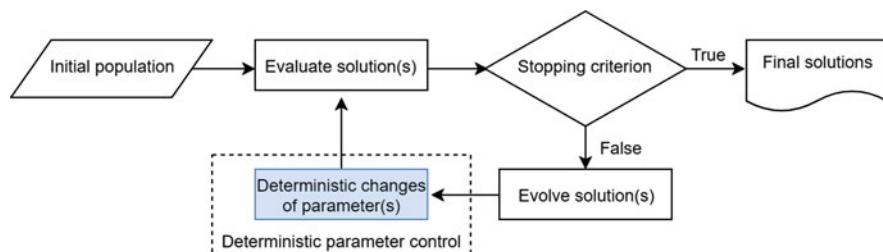


Fig. 11.6 A conceptual model of deterministic parameter control in EAs. The updates are following a predefined pattern, with no feedback from the optimisation process

11.4.1 *Deterministic Parameter Control*

In case of deterministic parameter control (see Fig. 11.6), the updates follow some predefined pattern and have no feedback from the optimisation process. Changing of the control parameters is therefore based on the elapsed time, measured in terms of the number of generations or the number of fitness evaluations or the wall-clock time, etc. The update rule is determined before the algorithm run; thus, finding the optimal deterministic update rules requires their tuning. The disadvantage of the static control parameter values is bypassed with this deterministic approach, but the algorithm is not able to identify the good control parameter values by itself.

Below some representative examples of deterministic parameter control are presented:

- Rule for the mutation strength of a GA, based on the dimension of the configuration space and the population size [38].
- Size of the actual population is changed (increased or decreased) every N fitness evaluation [39].
- Mutation rate changes in every iteration [40].
- Linear decrease of population size with occasional re-initialisation of the population size [41].
- Changing population sizes through specific wave function (i.e. inverse saw-tooth function) [42].
- In every iteration, random step size is used for a multi-valued problem [43].

11.4.2 *Adaptive Parameter Control*

During an optimisation process, various data are available that can give clues about the characteristics of the fitness landscape. This data can be used to guide parameter control so that optimisation is more efficient, given the current landscape. This approach is called adaptive parameter control. Its main trait is the feedback from the

optimisation process that changes the parameters according to some pre-described rule.

The following list presents some of the adaptive algorithms:

- (1+1) EA_{α} —a multiplicative, comparison-based update rule to adjust the mutation probability of a (1+1) Evolutionary Algorithm [24].
- PLES—the control parameters (population size, recombination, mutation) are calculated during the optimisation process according to the progress (best value and standard deviation) of the search [12].
- CMA-ES—the shape of mutation distribution is generated according to a covariance matrix C , which is adapted during evolution [25].
- SHADE—the control parameters for are drawn from a distribution, shape of which is guided by the values of control parameters that previously produced high-quality candidates [44].

Model of Adaptive Parameter Control

Parameter control requires searching for optimal control parameter values during the run. The optimisation starts with suboptimal control parameter values that are adapted during the progress of the algorithm. The parameter control derives the best next control parameter value such that the influence on the performance of the algorithm is optimised.

In this section we refer to the model and its main conceptual steps of the adaptive parameter control strategy and their classifications, based on [11].

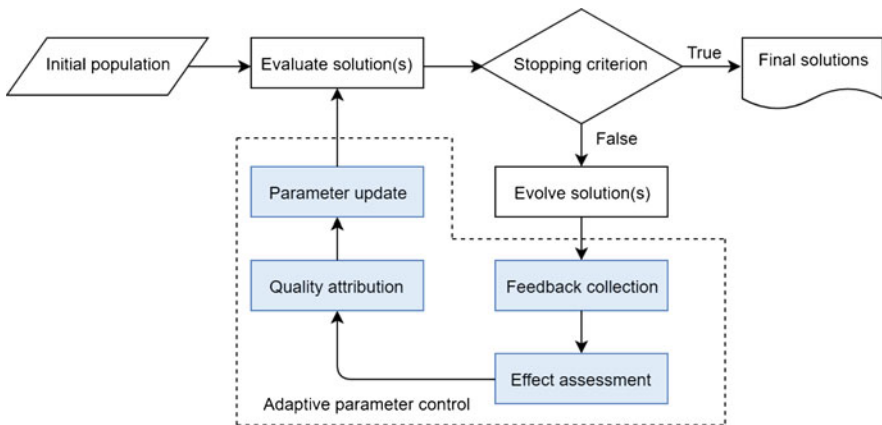


Fig. 11.7 A conceptual model of adaptive parameter control. The four main steps that are performed, namely (1) feedback collection, (2) effect assessment, (3) quality attribution and (4) parameter update, are represented as shaded boxes [11]

In Fig. 11.7 the conceptual model of this procedure is shown. During the optimisation, another iterative process is taking place that refers to the control parameters. The main steps of this process are (1) feedback collection, (2) effect assessment, (3) quality attribution and (4) parameter update. The success of each step depends on the others.

- Feedback collection: This is the starting step of the adaptive parameter control. This feedback measures certain properties of an EA, which indicate the algorithm's behaviour. According to [11], there are five different feedback categories:
 - Phenotype feedback: Feedback referring to the observed behaviour or quality of a solution. Most of the times, the fitness improvement of the solutions as an indication for the performance of the algorithm is used.
 - Relative phenotype feedback: Feedback referring to the relative quality of the solutions, most commonly using entropy as a measure.
 - Genotype feedback: This kind of feedback relates to the components of a single solution.
 - Genotype diversity feedback: Feedback as to how much population diversity is maintained for increasing search coverage and for dealing with fitness landscape change.
 - Feasibility feedback: Feedback about the amount of violation of constraints.
- Effect assessment: based on the feedback collection strategy, the effect of the parameter values on the performance of the algorithm is estimated. The distinction of current available parameter effect assessment methods is how the progress of the parameter values is defined, meaning what is evaluated as good performance for the algorithm. In Table 11.2 the different effects are depicted, depending on what solution is used as reference. For example, the *ancestor effect* uses as a reference to the solution of the parents and measures the improvement with the current solution, the *population effect* uses the solution of the population, etc.
- Quality attribution: Using the effect measured in previous iterations, a quality measure is defined to make a better choice of suitable control parameter value in the next iterations. According to what kind of change is taken into account, there exist the *immediate* which assumes that the change in the solution properties is directly related to the use of certain control parameter values, the *average* which

Table 11.2 List of effect assessments depending on the reference solution

Effect assessment	Reference solution
Ancestor effect	Solution of the parents
Population effect	Solution of the population
Best effect	Best solution of the population
Worst effect	Worst solution of the population
Median effect	Median solution of the population
Current effect	Current solution directly as effect (e.g. its fitness)

considers the average improvement in the properties of all the solutions in the population, the *extreme* which refers to outliers and *learned quality attribution* using other learning-based techniques, such as machine learning.

- Parameter update: The final step of adaptive parameter control is the update mechanism. The update of the parameters is a settlement between control parameter values with high quality and exploring new values. The categories are the following:
 - Quality proportionate: Each control parameter value is assigned with a probability, which defines how frequently it is used in future iterations.
 - Quality proportionate with minimum probability: The same as above, but each control parameter value has a minimum probability. That way, a value that is not performing well is not lost completely, as it can be successful in future iterations.
 - Greedy: The best control parameter value is selected to be used in the next iteration.
 - Deterministic: A rule depending on the current value is used to update the control parameter values of the next iteration.

Therefore, designing a good adaptive parameter control strategy requires some consideration. The difficulty of designing a good parameter control strategy is strongly connected to the problem of landscape analysis [45]. We can see parameter control as performing some sort of landscape analysis during optimisation, and on top of that, the features of the landscape are mapped to control parameter values that work well in such landscapes. One approach to circumvent these design difficulties is to let a machine learning model guide the parameter control [46]. By this method only the feedback collection step needs to be designed, whilst other steps are left to the machine learning model. Using a sufficient amount of data, the model could learn good rules for parameter control and in essence design remaining steps of parameter control based on which rules worked well during training of the model.

Reinforcement learning is especially suitable to find such rules for optimisation parameter control. This type of machine learning was found to be extremely successful to construct agents that are capable of playing various games [47]. In this regard, optimisation parameter control can be viewed as an agent playing a game. In the setting of parameter control being a game, a player performs actions (propose new values of optimisation parameters) that influence the behaviour of the optimisation process. At the end, the score of the player's actions is equivalent to the fitness found by the optimisation process. In this way, Zhang and Lu devised an evolutionary algorithm that uses a mutation operator proposed by a machine learning model trained using reinforcement learning [48]. However, this type of parameter control construction is still in its infancy and is currently an active area of research.

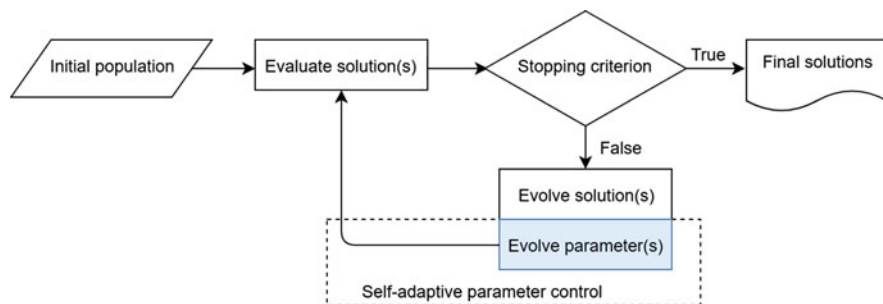


Fig. 11.8 A conceptual model of self-adaptive parameter control. The parameters are co-evolving with the solutions of the problem

11.4.3 Self-Adapting Parameter Control

A search for good parameter values for an optimisation process is by itself an optimisation problem. On the other hand, an optimisation algorithm that is being tuned can be used to find good parameter values. A given optimisation algorithm is designed to perform optimisation well and why not use it to find its own control parameter values. Self-adjusting parameter control is a method where the search for good control parameter values is performed simultaneously with the search of optimal problem values, by encoding the control parameters into the chromosome (as additional variables) (see Fig. 11.8).

In the case of EA, self-adjusting parameter control can be achieved by including the control parameters in the chromosome alongside the decision variables. In this way, an EA is able to try various control parameters and, through variation and selection, choose control parameter values that are the most appropriate in a given stage of optimisation. If some control parameter values lead the optimisation process to candidate solutions with higher fitness, these control parameter values are more likely to be inherited in the next generation. Control parameters that can be adjusted in such a way include the type of evolutionary operators used and parameters that determine their behaviour. For example, when using Gaussian mutation, the chromosome can include the individual's probability for mutation and the variance of mutation [49].

One disadvantage of self-adjusting parameter control in EA is that the inclusion of control parameters in the chromosome increases its size. This means that chromosome modifications become computationally more expensive [50]. One way to circumvent this issue is to use two populations. One population has individuals with chromosomes consisting of decision variables, whilst the other population has individuals with chromosomes consisting of control parameter values. This reduces the complexity of the optimisation by splitting the search space into two smaller search spaces.

Like the adaptive control, the self-adjusting control also uses feedback from the optimisation and searches for parameter values that are optimal for the region of

decision space that the population occupies. In that regard, this control adjusts the control parameters that are suitable for the current landscape. Consequently, this makes such an optimisation dynamic (see Sect. 11.5.2), and the optimal values of control parameters change during optimisation process.

The following list presents some of the self-adaptive implementations:

- Self-adaptation mechanism of mutation rates in GA [51]
- jDE—self-adaptive control parameter settings of differential evolution (DE) algorithm [26, 52]
- Self-adaptive refinement of mutation rates in EA [10]
- Self-adaptive mutation rate in the $(1,\lambda)$ EA [53]

11.4.4 Tuning vs. Control

Deciding whether to perform parameter tuning or to use some type of parameter control or to use some combination of both approaches (see combinations in Fig. 11.9) depends on the problem class and on how often optimisation problems need to be solved. Both parameter tuning and parameter control have their own advantages and drawbacks.

Parameter tuning can take a considerable amount of time, especially when there are many control parameters. Obtaining good control parameter values requires the use of an optimisation method that searches the values for which the algorithm has statistically the best performance. This performance measure of the optimisation algorithm is typically subjected to noise, and several runs need to be executed for the same control parameter values. The landscape of parameter tuning optimisation is usually non-separable and multi-dimensional. The optimal parameters can typically not be found in a sequential fashion, because of their complex interactions.

Fig. 11.9 Different combinations of using parameter tuning and parameter control [8]

		Control	
		YES	NO
Tuning	YES	Control mechanism tailored to application	Static values obtained while tuning
	NO	Control mechanism used out of the box	Static values based on intuition or convention

By performing a proper parameter tuning, we get values for control parameters that are good for the optimisation problems chosen in the tuning process. To reduce the computational burden of tuning, simpler, synthetic problems are often used that might not reflect the characteristics of real-world problems for which the tuned algorithm will be used. It was also shown that good control parameter values for one problem can behave poorly on similarly looking problems [8]. This is connected to the fact that the optimisation algorithm performance is usually quite sensitive with respect to control parameter values. A small change in control parameter value can result in great changes in algorithm overall performance.

A potential advantage of the static approach is that the process of parameter tuning is pre-described and theoretically well founded since tuning is an optimisation problem. From a design point of view, this is not the case for parameter control. Searching for an optimal control strategy can be considered an optimisation problem in infinite dimensional search space. Therefore, how to develop suitable update rules for parameter control is theoretically still an open question. Nevertheless, optimisation parameter control is a very active area of research that brought great performance improvements to several optimisation algorithms [11, 13].

Parameter control can also introduce additional parameters on its own. Such parameters determine how parameter control behaves. Therefore, the use of parameter control does not necessarily bring parameterless optimisation algorithms. However, parameter control rules should be designed in a way that the additional parameters (so called meta-parameters) of the parameter control are much less sensitive compared to original parameters of the optimisation algorithm [4, 54].

An advantage of the non-static parameter approach is the gain of flexibility and the possibility to adjust the parameter values to the current state of the search process. Since characteristics of the fitness landscape change during optimisation, it is much more efficient to change parameters during optimisation and use appropriate control parameter values in each stage of the optimisation. Another advantage of parameter control is that it can reduce the need for parameter tuning. If adaptive or self-adaptive parameter control is used, at the first stage of optimisation, the algorithm discovers the proper parameters itself, without tuning. This can slightly prolong the optimisation, but it can eliminate the need for time-consuming parameter tuning.

The choice of whether to use parameter tuning, or some combination of tuning, and control (see Fig. 11.9), depends on the complexity of the problems and in what frequency these particular optimisation problems are solved. If the problem is very hard and needs to be solved only once, parameter tuning represents an enormous overhead. If a hard problem needs to be solved over and over again, parameter tuning can represent a negligible overhead and can reduce the optimisation time considerably. In some cases it might be useful to first search for fine-tuned initial values and then to use parameter control. This way the tuning gives the advantage of tailoring to a specific problem class and the control offers the benefits of dynamically varying parameter values [8].

11.5 Real-World Optimisation

Real-world optimisation problems occur in many applications such as engineering design, scientific modelling, image processing, etc. In general, these problems contain non-linear objective functions of mixed design variables i.e. continuous and discrete, with linear as well as non-linear constraints. These problems might have several local optima, causing trouble to heuristic methods, as these methods do not guarantee to locate the global optimum. For a wide range of real-world optimisation problems, though, a near-optimal or a better-than-known solution is considered a satisfactory result of an optimisation problem.

Real-world systems are, in general, large and very complex. They require to process a large amount of data, to perform complex optimisation and make decisions fast [55]. Real-world optimisation problems consist of several characteristics that increase the complexity of the optimum solution search. Some of the characteristics of the problems for which parameter control in optimisation could appear advantageous are the following:

- Number and type of variables: Large number of decision variables, with problems that are known as large-scale global optimisation (LSGO) problems [56]. Also, mixed-integer problems, where different types of variables are optimised.
- Dynamic problems: Problems that are changing over time.
- Problems under uncertainty: The variables of the problem have some uncertainty.
- Number of objectives: Problems that require optimising more than one objective function simultaneously and need to be solved by a multi/many-objective approach.
- Nested problems: Multi-/bilevel optimisation, where one optimisation problem has another optimisation problem as a constraint.

As mentioned before properly defined control parameters play a crucial role in effectively handling the above characteristics and solving such problems. For example, with increasing dimensionality of the problem, its landscape complexity grows and the search space increases exponentially. However, an optimisation algorithm must be able to explore the entire search space efficiently.

11.5.1 Large-Scale Global Optimisation

LSGO, where the problem dimension D (the number of variables to be optimised) has an order of magnitude of around $D = 1000$, is an active research field due to the growing number of large-scale optimisation problems in engineering, manufacturing and economy applications (such as bio-computing, data or web mining, scheduling, vehicle routing, etc.) [57, 58]. Most engineering problems have an exponential increase in the number of required decision variables [59]. Advances in machine learning and the wide use of deep artificial neural networks

result in optimisation problems with over a billion variables [60]. Large-scale optimisation is also present with many data analytic and learning problems [61]. The problems range from shape design optimisation for aircraft wings and turbine blades [62], satellite layout design [63] and parameter calibration of water distribution system [64] to seismic waveform inversion [65]. A major challenge of large-scale optimisation is the exponential growth in the size of the search space with respect to the number of decision variables [56].

In recent years, LSGO gained attention and attracted wide interest from researchers and practitioners as well as mathematicians and engineers. The challenges motivated the design of many kinds of efficient, effective and robust kinds of metaheuristic algorithms to solve LSGO problems with high-quality solutions and high convergence performance as well as with low computational cost [66].

To achieve acceptable results even for the same problem, different parameter settings along with different reproduction schemes at different stages of optimisation process are needed. Therefore, several techniques (e.g. [67, 68]) have been designed to adjust control parameters in an adaptive or self-adaptive manner instead of a trial-and-error procedure.

11.5.2 *Dynamic Optimisation*

Real-world optimisation problems are usually subject to changing conditions over time. The effects of these changes could influence several aspects of the problem, such as the objective function, the problem instance, its constraints etc. Therefore, the optimal solution to the problem might change over time. These problems, when solved by an optimisation algorithm on-the-fly, are called dynamic optimisation problems (DOPs) [69].

One can understand that optimising dynamic problems is not a simple task. The algorithm is expected to be able to track the current optimal solution as well as the changing optimal solution over time. Therefore, the optimisation procedure has to be able to detect these changes and react quick enough. This also requires dynamic change of the ratio for exploration and exploitation parts of the search. Both adaptive [70] and self-adaptive [71] parameter control can be used.

Based on a comprehensive survey [72], four different strategies can be used to help population-based algorithms to adapt in dynamical environments:

- Increasing diversity of the population after a change is detected, e.g. by increasing mutation rate every N generations
- Maintaining diversity throughout the run, to avoid convergence of the population on one point
- Memory-based approaches, taking into consideration older solutions and sometimes making predictions based on historical data

- Multi-population approaches, where many small populations track their own peaks as the environment changes

The above strategies require a parameter control approach.

11.5.3 *Optimisation Under Uncertainty*

The presence of a range of uncertainties has to be taken into account for solving many real-world applications with EAs. Jin and Branke [73] categorise the uncertainties that influence EA performance into four types:

- When noise occurs in the fitness function
- In case of design and environmental parameters subject to changes after the optimisation
- When fitness function is an approximation
- When the optimum changes over time (as in dynamic optimisation, see Sect. 11.5.2)

Methodologies to addressing noisy fitness function are *explicit averaging* by calculating the average of the fitness values over a number of randomly sampled disturbances (see, e.g., [74, 75]), *implicit averaging* sample size as an inverse function of the population size [76], *fitness inheritance* where the offspring inherits also the mean and standard deviation of the objective value [77] and *selection modification* [78]. These methods assume that the search space follows a homogeneous noise distribution such as a uniform or a normal distribution [79].

11.5.4 *Multi-objective Optimisation*

Multi-objective and also many-objective optimisation (see details in Chap. 8) approaches are used for optimisation problems where several criteria need to be optimised that are equally treated and not merged (e.g. by weights) into one single objective. The output of multi-objective optimisation is a set of solutions that approximates the Pareto front. The main difference here is that there is no unique measure that would indicate how good a current approximation of the Pareto front is. Unlike in single-objective case where fitness can be used to measure this. Therefore, in multi-objective case, adaptive parameter control is a bit more complicated to design and additional considerations are needed to design the phenotype feedback collection part.

One possibility for assessing in what stage the optimisation process is, is to monitor the proportion of non-dominated solutions in the population [80], or convergence detection [81]. This quantity typically increases during optimisation process and can be used to guide the parameter control. The most common

indicators that are also used as input to parameter control are the crowding distance and the contributing hypervolume [82, 83]. Other metrics can also be applied such as ε -dominance, generational distance, delta indicator, two-set coverage and so on [84]. Compared to adaptive control, self-adaptive control is easier to design and implement because less modifications are needed to upgrade an existing multi-objective optimisation algorithm [85, 86].

In general, the multi-objective optimisation approach is able to find high-quality solutions by adapting to specific dynamic conditions through selection of appropriate solutions from Pareto front [55]. To further improve its performance, the control parameter values should also be adapted to lead the search through the changing system.

11.5.5 *Multilevel Optimisation*

In many real-world processes, there is a hierarchy of decision-makers and therefore decisions are taken at different levels [87]. The constraint domain associated with a multilevel problem is implicitly determined by a series of optimisation problems that must be solved in a predetermined sequence. The simplest form of a multilevel problem is the one with two levels, called bilevel optimisation problem. The optimisation of such problem aims to achieve the optimum solution of the upper (in hierarchy) level, whilst the optimum of the lower optimisation level is also taken into account. This forms a challenging complex problem, as for every upper-level vector, a whole optimisation task of the lower level problem is required. This usually leads to a problem that is non-linear and non-convex and in general does not follow any simplified assumptions. Since the lower-level landscape changes for every upper-level vector, parameter control when using EAs to tackle with this problem can be useful. More details about multi-/bilevel optimisation and how it is solved can be found in Chap. 9.

Another very interesting application of bilevel optimisation that was developed recently and is connected to parameter tuning was formulating the parameter tuning of EAs as bilevel optimisation problem. In [88] the authors proposed the parameter tuning problem as an inherently bilevel programming problem involving algorithmic performance as the objective(s), introducing an evolutionary bilevel algorithm for parameter tuning. They tested it to few commonly used optimisation algorithms (differential evolution and Nelder-Mead) and it was found to obtain a fast convergence to the most efficient control parameter values. In the same vein, [89] created a bilevel framework for parallel tuning of optimisation control parameters and compared it to irace proving that it can be competitive. Bilevel control parameter tuning can be used to design a parameter control mechanism [90].

11.6 Summary

This chapter provides an insight into parameter control strategies in EA. The concept of evolutionary algorithms is introduced, along with the control parameters that need to be set for this kind of algorithms, their mutual interactions and their influence on the algorithm performance. Parameter tuning and parameter control, the two categories of parameter settings, are then explained. More focus is given to the three strategies of parameter control, i.e. deterministic, adaptive and self-adaptive parameter control. It also discusses when and why it is more appropriate to use either parameter tuning or parameter control strategies. Last but not least, it outlines the motivation of using parameter control for solving some instances of real-world optimisation problems, like the ones with a large number of variables, dynamic changes, multiple objectives, multiple levels and uncertainty.

Acknowledgments This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, grant agreement number 722734, and through the SYNERGY Twinning project, H2020-TWINN-2015, grant agreement number 692286. The authors acknowledge the financial support from the Slovenian Research Agency (research core funding no. P2-0098 and PR-07606).

References

1. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn. (Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1989)
2. X.-S. Yang, Review of meta-heuristics and generalised evolutionary walk algorithm. *Int. J. Bio-Inspired Comput.* **3**(2), 77–84 (2011)
3. E.-G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 74 (Wiley, New York, 2009)
4. A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*, 2nd edn. (Springer Publishing Company, Incorporated, Berlin, 2015)
5. F.F. Orsucci, N. Sala, *Reflexing Interfaces: The Complex Coevolution of Information Technology Ecosystems*. IGI Global, January 2001 [Online]. Available <https://www.igi-global.com/book/reflexing-interfaces-complex-coevolution-information/865>
6. A.E. Eiben, C.A. Schippers, On evolutionary exploration and exploitation. *Fundam. Inf.* **35**(1–4), 35–50 (1998) [Online]. Available <http://dl.acm.org/citation.cfm?id=297119.297124>
7. L. Hong, J.H. Drake, J.R. Woodward, E. Özcan, A hyper-heuristic approach to automated generation of mutation operators for evolutionary programming. *Appl. Soft Comput.* **62**, 162–175 (2018) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S1568494617306051>
8. G. Karafotias, M. Hoogendoorn, A.E. Eiben, Parameter control in evolutionary algorithms: trends and challenges. *IEEE Trans. Evol. Comput.* **19**(2), 167–187 (2015)
9. A.E. Eiben, S.K. Smit, Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol. Comput.* **1**(1), 19–31 (2011)
10. D.-C. Dang, P.K. Lehre, Self-adaptation of mutation rates in non-elitist populations, in *Parallel Problem Solving from Nature – PPSN XIV*, ed. by J. Handl, E. Hart, P.R. Lewis, M. López-Ibáñez, G. Ochoa, B. Paechter (Springer International Publishing, Cham, 2016), pp. 803–813

11. A. Aleti, I. Moser, A systematic literature review of adaptive parameter control methods for evolutionary algorithms. *ACM Comput. Surv.* **49**(3), 56:1–56:35 (2016) [Online]. Available <http://doi.acm.org/10.1145/2996355>
12. G. Papa, Parameter-less algorithm for evolutionary-based optimization. *Comput. Optim. Appl.* **56**(1), 209–229 (2013) [Online]. Available <https://link.springer.com/article/10.1007/s10589-013-9565-4>
13. C. Doerr, Dynamic parameter choices in evolutionary computation, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '18* (ACM, New York, NY, 2018), pp. 800–830 [Online]. Available <http://doi.acm.org/10.1145/3205651.3207851>
14. A.E. Eiben, S.K. Smit, *Evolutionary Algorithm Parameters and Methods to Tune Them* (Springer, Berlin, Heidelberg, 2012), pp. 15–36 [Online]. Available https://doi.org/10.1007/978-3-642-21434-9_2
15. P. Machado, A. Leitão, Evolving fitness functions for mating selection, in *Genetic Programming*, ed. by S. Silva, J.A. Foster, M. Nicolau, P. Machado, M. Giacobini (Springer, Berlin, Heidelberg, 2011), pp. 227–238
16. C.M. Fonseca, M.B. Correia, Developing redundant binary representations for genetic search, in *2005 IEEE Congress on Evolutionary Computation*, September 2005, vol. 2, pp. 1675–1682
17. E.K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, J.R. Woodward, *A Classification of Hyper-heuristic Approaches* (Springer US, Boston, MA, 2010), pp. 449–468 [Online]. Available https://doi.org/10.1007/978-1-4419-1665-5_15
18. G. Papa, J. Šilc, Evolutionary synthesis algorithm - genetic operators tuning, in *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation* (WSEAS Press, Athens, 2002), pp. 256–261
19. A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* **3**(2), 124–141 (1999)
20. I. Rechenberg, *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution* (Frommann-Holzboog, Stuttgart, 1973)
21. D.E. Goldberg, Messy genetic algorithms revisited: studies in mixed size and scale. *Complex Syst.* **4**(4), 15–444 (1990)
22. T.-L. Yu, D.E. Goldberg, Toward an understanding of the quality and efficiency of model building for genetic algorithms, in *Genetic and Evolutionary Computation—GECCO 2004*, vol. 3103, ed. by T. Kanade, J. Kittler, J.M. Kleinberg, F. Mattern, J.C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M.Y. Vardi, G. Weikum, K. Deb (Springer, Berlin, Heidelberg, 2004), pp. 367–378
23. T. Bäck, Optimal mutation rates in genetic search, in *Proceedings of the fifth International Conference on Genetic Algorithms* (Morgan Kaufmann, San Mateo, CA, 1993), pp. 2–8
24. C. Doerr, M. Wagner, Simple on-the-fly parameter selection mechanisms for two classical discrete black-box optimization benchmark problems, in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018*, Kyoto, July 15–19, 2018, pp. 943–950 [Online]. Available <https://doi.org/10.1145/3205455.3205560>
25. N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**(2), 159–195, June 2001 [Online]. Available: <http://dx.doi.org/10.1162/106365601750190398>
26. J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10**(6), 646–657 (2006)
27. K.A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, Ph.D. dissertation, University of Michigan, Ann Arbor, MI, 1975
28. J.J. Grefenstette, Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybern.* **16**(1), 122–128 (1986)

29. H. Muhlenbein, How genetic algorithms really work: I. Mutation and hillclimbing, in *Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature, 1992, 1992* [Online]. Available: <https://ci.nii.ac.jp/naid/10022158367/en/>
30. H.-G. Beyer, B. Sendhoff, Covariance matrix adaptation revisited – the cma evolution strategy –, in *Parallel Problem Solving from Nature – PPSN X*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, C. Poloni (Springer, Berlin, Heidelberg, 2008), pp. 123–132
31. M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, T. Stützle, M. Birattari, The irace package: iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **3**, 43–58 (2016)
32. T. Bartz-Beielstein, O. Flasch, P. Koch, W. Konen, SPOT: a toolbox for interactive and automatic tuning in the R environment, in *Proceedings 20. Workshop Computational Intelligence*, ed. by F. Hoffmann, E. Hüllermeier (Universitätsverlag Karlsruhe, Karlsruhe 2010), pp. 264–273
33. C. Ansótegui, Y. Malitsky, H. Samulowitz, M. Sellmann, K. Tierney, Model-based genetic algorithms for algorithm configuration, in *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15* (AAAI Press, Menlo Park, CA, 2015), pp. 733–739 [Online]. Available <http://dl.acm.org/citation.cfm?id=2832249.2832351>
34. F. Hutter, H.H. Hoos, K. Leyton-Brown, T. Stützle, Paramils: an automatic algorithm configuration framework. *J. Artif. Intell. Res.* **36**(1), 267–306 (2009) [Online]. Available <http://dl.acm.org/citation.cfm?id=1734953.1734959>
35. F. Hutter, H.H. Hoos, K. Leyton-Brown, Sequential model-based optimization for general algorithm configuration, in *Proceedings of the 5th International Conference on Learning and Intelligent Optimization, LION'05* (Springer, Berlin, Heidelberg, 2011), pp. 507–523 [Online]. Available http://dx.doi.org/10.1007/978-3-642-25566-3_40
36. J. Snoek, H. Larochelle, R.P. Adams, Practical Bayesian optimization of machine learning algorithms, in *Advances in Neural Information Processing Systems* (2012), pp. 2951–2959
37. A.E. Eiben, J. Smith, From evolutionary computation to the evolution of things. *Nature* **521**(7553), 476–482 (2015) [Online]. Available <https://www.nature.com/articles/nature14544>
38. J. Hesser, R. Männer, Towards an optimal mutation probability for genetic algorithms, in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature, PPSN I* (Springer, Berlin, Heidelberg, 1991), pp. 23–32 [Online]. Available <http://dl.acm.org/citation.cfm?id=645821.670199>
39. J.C. Costa, R. Tavares, A. Rosa, An experimental study on dynamic random variation of population size, in *IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics*, October 1999, vol. 1, pp. 607–612
40. T. Jansen, I. Wegener, On the analysis of a dynamic evolutionary algorithm. *J. Discrete Algorith.* **4**(1), 181–199 (2006) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S1570866705000109>
41. V.K. Koumousis, C.P. Katsaras, A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Trans. Evol. Comput.* **10**(1), 19–28 (2006)
42. T. Hu, S. Harding, W. Banzhaf, Variable population size and evolution acceleration: a case study with a parallel evolutionary algorithm. *Genet. Program. Evol. Mach.* **11**(2), 205–225 (2010) [Online]. Available <http://dx.doi.org/10.1007/s10710-010-9105-2>
43. B. Doerr, C. Doerr, Optimal static and self-adjusting parameter choices for the $(1+(\lambda,\lambda))$ genetic algorithm. *Algorithmica* **80**(5), 1658–1709 (2018) [Online]. Available <https://doi.org/10.1007/s00453-017-0354-9>
44. R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in *IEEE Congress on Evolutionary Computation (CEC)* (IEEE, New York, 2013), pp. 71–78
45. P. Merz, Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput.* **12**(3), 303–325 (2004)
46. J. Zhang, Z.-H. Zhan, Y. Lin, N. Chen, Y.-J. Gong, J.-H. Zhong, H.S. Chung, Y. Li, Y.-H. Shi, Evolutionary computation meets machine learning: a survey. *IEEE Comput. Intell. Mag.* **6**(4), 68–75 (2011)

47. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning (2013). Preprint. arXiv:1312.5602
48. H. Zhang, J. Lu, Adaptive evolutionary programming based on reinforcement learning. *Inf. Sci.* **178**(4), 971–984 (2008)
49. X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**(2), 82–102 (1999)
50. O. Kramer, Evolutionary self-adaptation: a survey of operators and strategy parameters. *Evol. Intell.* **3**(2), 51–65 (2010)
51. T. Bäck, The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm, in *Proceedings of the 2nd Conference on Parallel Problem Solving from Nature, PPSN 1992*, ed. by R. Männer, B. Manderick (North-Holland, Amsterdam, 1992)
52. J. Brest, M.S. Maučec, Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft Comput.* **15**(11), 2157–2174 (2011) [Online]. Available <https://doi.org/10.1007/s00500-010-0644-5>
53. B. Doerr, C. Witt, J. Yang, Runtime analysis for self-adaptive mutation rates, in *Proceedings of the Genetic and Evolutionary Computation Conference, ECCO '18* (ACM, New York, NY, 2018), pp. 1475–1482 [Online]. Available <http://doi.acm.org/10.1145/3205455.3205569>
54. P. Siarry, Z. Michalewicz, *Advances in Metaheuristics for Hard Optimization*. Natural Computing Series, 1st edn. (Springer, New York, 2007)
55. P. Korošec, U. Bole, G. Papa, A multi-objective approach to the application of real-world production scheduling. *Exp. Syst. Appl.* **40**(15), 5839–5853 (2013) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0957417413003321>
56. M.N. Omidvar, M. Yang, Y. Mei, X. Li, X. Yao, DG2: a faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Trans. Evol. Comput.* **21**(6), 929–942 (2017)
57. D.M. Cabrera, Evolutionary algorithms for large-scale global optimisation: a snapshot, trends and challenges. *Progr. Artif. Intell.* **5**(2), 85–89 (2016) [Online]. Available <https://doi.org/10.1007/s13748-016-0082-4>
58. X. Li, K. Tang, Z. Suganthan, P.N. Yang, Editorial for the special issue of information sciences journal (ISJ) on “Nature-inspired algorithms for large scale global optimization”. *Inf. Sci.* **316**, 437–439 (2015)
59. G. Vanderplaats, *Very Large Scale Optimization* (National Aeronautics and Space Administration, Washington, DC, 2002)
60. G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
61. Z.H. Zhou, N.V. Chawla, Y. Jin, G.J. Williams, Big data opportunities and challenges: discussions from data analytics perspectives [discussion forum]. *IEEE Comput. Intell. Mag.* **9**(4), 62–74 (2014)
62. Z. Yang, B. Sendhoff, K. Tang, X. Yao, Target shape design optimization by evolving b-splines with cooperative coevolution. *Appl. Soft Comput.* **48**(C), 672–682 (2016) [Online]. Available <https://doi.org/10.1016/j.asoc.2016.07.027>
63. H.-F. Teng, Y. Chen, W. Zeng, Y.-J. Shi, Q.-H. Hu, A dual-system variable-grain cooperative coevolutionary algorithm: satellite-module layout design. *IEEE Trans. Evol. Comput.* **14**(3), 438–455 (2010) [Online]. Available <http://dx.doi.org/10.1109/TEVC.2009.2033585>
64. Y. Wang, J. Huang, W.S. Dong, J.C. Yan, C.H. Tian, M. Li, W.T. Mo, Two-stage based ensemble optimization framework for large-scale global optimization. *Eur. J. Oper. Res.* **228**(2), 308–320 (2013) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0377221712009691>
65. C. Wang, J. Gao, High-dimensional waveform inversion with cooperative coevolutionary differential evolution algorithm. *IEEE Geosci. Remote Sens. Lett.* **9**(2), 297–301 (2012)
66. A.W. Mohamed, A.S. Almazayad, Differential evolution with novel mutation and adaptive crossover strategies for solving large scale global optimization problems. *Appl. Comput. Intell. Soft Comput.* **2017**, 1–18 (2017)

67. A. Zamuda, J. Brest, B. Bošković, V. Žumer, Large scale global optimization using differential evolution with self-adaptation and cooperative co-evolution, in *IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, June 2008, pp. 3718–3725
68. S. Das, S.S. Mullick, P. Suganthan, Recent advances in differential evolution—an updated survey. *Swarm Evol. Comput.* **27**, 1–30 (2016) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S2210650216000146>
69. T.T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: a survey of the state of the art. *Swarm Evol. Comput.* **6**, 1–24 (2012) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S2210650212000363>
70. H. Wang, D. Wang, S. Yang, A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Comput.* **13**(8–9), 763–780 (2009)
71. J. Brest, A. Zamuda, B. Bošković, M. S. Maučec, V. Žumer, Dynamic optimization using self-adaptive differential evolution, in *IEEE Congress on Evolutionary Computation, 2009. CEC'09* (IEEE, New York, 2009), pp. 415–422
72. J. Branke, Evolutionary approaches to dynamic optimization problems - updated survey, in *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems* (2001), pp. 27–30
73. Y. Jin, J. Branke, Evolutionary optimization in uncertain environments—a survey. *IEEE Trans. Evol. Comput.* **9**(3), 303–317 (2005)
74. H. Greiner, Robust optical coating design with evolutionary strategies. *Appl. Opt.* **35**(28), 5477–5483 (1996) [Online]. Available <http://ao.osa.org/abstract.cfm?URI=ao-35-28-5477>
75. B.L. Miller, Noise, sampling, and efficient genetic algorithms, Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1997
76. J.M. Fitzpatrick, J.J. Grefenstette, Genetic algorithms in noisy environments. *Mach. Learn.* **3**(2–3), 101–120 (1988)
77. L.T. Bui, H.A. Abbass, D. Essam, Fitness inheritance for noisy evolutionary multi-objective optimization, in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation* (ACM, New York, 2005), pp. 779–785
78. J. Teich, Pareto-front exploration with uncertain objectives, in *International Conference on Evolutionary Multi-Criterion Optimization* (Springer, New York, 2001), pp. 314–328
79. M. Vallejo, D.W. Corne, Evolutionary algorithms under noise and uncertainty: a location-allocation case study, in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (2016), pp. 1–10
80. D. Yang, L. Jiao, M. Gong, Adaptive multi-objective optimization based on nondominated solutions. *Computat. Intell.* **25**(2), 84–108 (2009)
81. B. Naujoks, H. Trautmann, Online convergence detection for multiobjective aerodynamic applications, in *2009 IEEE Congress on Evolutionary Computation*, May 2009, pp. 332–339
82. C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.* **15**(1), 1–28 (2007)
83. N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Oper. Res.* **181**(3), 1653–1669 (2007) [Online]. Available <http://www.sciencedirect.com/science/article/pii/S0377221706005443>
84. N. Riquelme, C. Von Lücken, B. Baran, Performance metrics in multi-objective optimization, in *Computing Conference (CLEI), 2015 Latin American* (IEEE, New York, 2015), pp. 1–11
85. Y.-N. Wang, L.-H. Wu, X.-F. Yuan, Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Comput.* **14**(3), 193–209 (2010)
86. R. Cao, G. Li, Y. Wu, A self-adaptive evolutionary algorithm for multi-objective optimization, in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, ed. by D.-S. Huang, L. Heutte, M. Loog (Springer, Berlin, Heidelberg, 2007), pp. 553–564

87. A. Migdalas, P.M. Pardalos, P. Vrbrand, *Multilevel Optimization: Algorithms and Applications*, 1st edn. (Springer Publishing Company, Incorporated, New York, 2012)
88. A. Sinha, P. Malo, P. Xu, K. Deb, A bilevel optimization approach to automated parameter tuning, in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO '14* (ACM, New York, NY, 2014), pp. 847–854 [Online]. Available <http://doi.acm.org/10.1145/2576768.2598221>
89. M. Andersson, A Bilevel Approach to Parameter Tuning of Optimization Algorithms Using Evolutionary Computing, Ph.D. dissertation, University of Skövde, 2018.
90. M. Andersson, S. Bandaru, A. Ng, A. Syberfeldt, Parameter tuning of MOEAs using a bilevel optimization approach, in *Evolutionary Multi-Criterion Optimization*, ed. by A. Gaspar-Cunha, C. Henggeler Antunes, C.C. Coello (Springer International Publishing, Cham, 2015), pp. 233–247

Chapter 12

Response Surface Methodology



Péter Zénó Korondi , Mariapia Marchi , and Carlo Poloni 

Abstract Response Surface Methods (RSMs) are statistical and numerical models that approximate the relationship between multiple input variables and an output variable. This chapter introduces the methodology and its importance for engineering design optimisation. The basic steps to build RSMs and validate the model accuracy are explained. An overview of three classical methods (Least Squares, Radial Basis Functions, and Kriging) is provided. A simple wing structure design optimisation problem is used to illustrate the different phases of the response surface methodology and its application to design optimisation. This example also includes the case of noisy data.

Keywords Response surface method · Radial basis function · Kriging · Surrogate model · Quality indicators · Design optimisation

12.1 Introduction

Response Surface Methodology refers to statistical and numerical techniques to model the relationship between multiple input variables and an output variable. A Response Surface Method (RSM) can be considered as a multidimensional surface

P. Z. Korondi (✉)
ESTECO SpA, Trieste, Italy
Department of Engineering and Architecture, University of Trieste, Trieste, Italy
e-mail: korondi@esteco.com

M. Marchi
ESTECO SpA, Trieste, Italy
e-mail: marchi@esteco.com

C. Poloni
Department of Engineering and Architecture, University of Trieste, Trieste, Italy
ESTECO SpA, Trieste, Italy
e-mail: poloni@units.it; poloni@esteco.com

fitting of the output variable based on the observed data in multidimensional input space.

Generally speaking, a response surface model (a.k.a. surrogate model or meta-model) is used to replace expensive numerical or physical experiments with a computationally cheap and sufficiently accurate model. In engineering, decisions are made on information obtained from various kinds of analyses, as pointed out by Forrester in his book [1]. One way to get information and increase the knowledge of a problem is to conduct experiments; however, in many cases the cost and complexity of the experiments is so high that only a limited number, if any, of observations is feasible. For example, in aerospace engineering, experiments can be very expensive (e.g. extra-territorial missions) or can take a long time (e.g. run high-fidelity simulations). Consequently, models are built to increase the knowledge gained from the observations and to predict performance values which cannot be directly observed.

RSMs are a valuable tool for making decisions during the life-cycle of a physical asset or process. In early-stage design phases, RSMs can give fast and cheap predictions about the cause–effect relationship of the design inputs and outputs. In Robust Design Optimisation and Reliability Based Design Optimisation, RSMs can be used to substitute expensive performance analyses to provide enough data for statistical calculations. In operational phases, RSMs can provide additional data, information for digital twins, operators and artificial intelligence algorithms to facilitate the control of the process. In recent decades, engineering design has become increasingly collaborative and multi-disciplinary. In many cases, the parameters of an expensive numerical experiment depend on the output of one or more other expensive numerical experiments of different disciplines. In this context, the RSM of a numerical experiment can provide response data for input parameter predictions of numerical solvers of other disciplines, facilitating the use of parallel discipline analyses [2]. Moreover, RSMs are able to handle possibly noisy experimental results.

The accuracy of RSMs can be controlled by the configuration of the RSM algorithm; however, speed and accuracy often conflict: and a compromise has to be made. The task of choosing the most appropriate RSM is challenging and requires expertise.

This chapter is organised as follows. Section 12.2 introduces the pragmatical stages and steps of the RSM construction. Section 12.3 provides a brief overview of some selected RSMs: Least Squares Method (LSM), Radial Basis Functions (RBF), and Kriging. Section 12.4 shows the application of the RSMs in an uncertainty analysis of a simplified wing design problem.

For a more detailed review on RSMs see one of the following books [1, 3, 4].

12.2 Response Surface Model Construction

12.2.1 Objective

The objective of RSMs is to map the behaviour of an output variable based on the known values of the input variables. In mathematical terms,

$$y = f(\mathbf{x}), \quad (12.1)$$

where \mathbf{x} is the vector of input variables, f is the mapping function, and y is the output variable.

In practice, the perfect map cannot be found, except in some simple problems. A RSM provides only an approximate map between the output and inputs:

$$\hat{y} = \hat{f}(\mathbf{x}), \quad (12.2)$$

where \hat{f} is the RSM and \hat{y} is the predicted output, which usually deviates from its real value,

$$y = \hat{y} + \epsilon, \quad (12.3)$$

where ϵ represents the error with respect to the real output value.

12.2.2 Classification

RSMs can be divided into two main categories: regression methods and interpolation methods.

Regression RSM (or approximation method) is typically a polynomial function or nonlinear model with coefficients that are set to minimise the error between the predicted surface and the sample points of the training data set. Approximation methods are particularly useful for modelling ill-conditioned problems or in case of low quality training data (e.g. noisy data sets).

Interpolating RSM passes through the points of the training data set. In other words, the interpolating surface and the real response surface are equal at the locations of the training data. Hence, interpolation is well-suited to deterministic computer experiments where the simulation error is negligible. The choice of inappropriate interpolation methods can result in oscillatory surfaces (e.g. when there is a large number of model parameters). Consequently, extrapolation based on an interpolated response surface with many model parameters is not suggested without additional validation of the extrapolated predictions.

12.2.3 Construction Stages

The construction of a response surface model can be divided into four stages: data preparation, algorithm choice, parameter tuning, and model validation. Each stage consists in multiple steps described in more detail in the following sections. The descriptions are not immutable. For example, the first stage can be skipped if a sufficient amount of data is available. The stages do not necessarily have to be performed sequentially. In numerous applications these stages are done iteratively. The training data can be refined in regions where the model validation phase indicates lower accuracy and models can be recalculated after the introduction of new sample instances.

12.2.3.1 Data Preparation

The first step is to identify the relevant design variables and responses based on available knowledge or sensitivity analysis [5]. The region of interest (i.e. the ranges of the design variables) should also be defined.

The second step consists in sampling the design space. The number of possible sample points is constrained by the cost of the experiments. Different techniques have been developed to obtain the maximum amount of information using the minimal amount of resources. These techniques are called Design of Experiments (DOE) [6]. Many DOE techniques solve an optimisation problem to achieve a well-distributed sample data set. It should not contain data points that are too close to each other or large areas not covered by any points. Two basic quality measures are introduced below and shown in Fig. 12.1: the Separation Distance index and Fill Distance index.

The Separation Distance q is the distance between the two closest training points and is defined as,

$$q = \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\|, \quad (12.4)$$

where \mathbf{x}_i and \mathbf{x}_j is any pair of sample points and the norm $\|\dots\|$ is the Euclidean distance in a multidimensional space. Training points that are too close to each other can lead to numerical instabilities in the generation of the RSM, hence, the Separation Distance should be maximised.

The Fill Distance index h is a metric to quantify how well the training points cover the investigated domain Ω of the variables. In case of a two-variable domain, the Fill Distance gives the radius of the biggest possible circle that can be drawn in the domain without having any training point inside the circle.

$$h = \max_{\mathbf{x} \in \Omega} \min_{1 \leq j \leq n} \|\mathbf{x} - \mathbf{x}_j\|. \quad (12.5)$$

Fig. 12.1 Separation distance q and fill distance h for a two-dimensional sample of training points

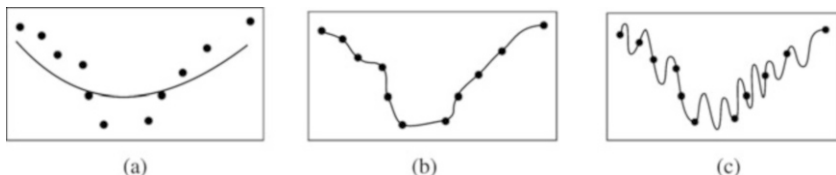
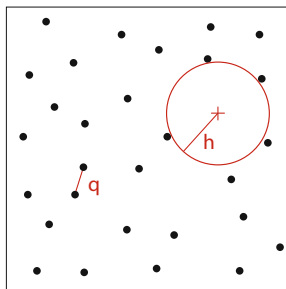


Fig. 12.2 Examples of model fitting to the training data set. (a) Underfitting. (b) Correct fitting. (c) Overfitting

The Fill Distance should be minimised in order to increase the quality of the RSM prediction. For example, a good quality training data set can be achieved by maximising the $\frac{q}{h}$ relationship.

12.2.3.2 Algorithm Choice

The challenge of finding the appropriate algorithm for predicting the true system response with an RSM (see Fig.12.2b) is a task that, given the great amount of possible RSMs, requires expertise. Consequently, physical insight into the investigated problem can help to find the most appropriate algorithm. For instance, the aerodynamic lift coefficient of an airfoil or an aircraft wing depends on the angle-of-attack ($a-o-a$) and can be approximated by a linear model for a certain regime of $a-o-a$ values. The assumption of such simple RSM, however, blurs away the fine details of the lift- $a-o-a$ relation and can lead to underfitting (see Fig.12.2a). In light of this, complex models with high flexibility are preferable as they can adapt to simple and complex surfaces, provided that the model parameters are chosen well. Inversely, having too many model parameters with respect to the training points can add significant spurious noise and can result in overfitting (see Fig.12.2c).

12.2.3.3 Model Training

In this stage, model parameters are tuned, or set, to fit the chosen RSM to the training data.

In interpolation, the model parameters are typically found by solving a linear system of equations defined by the interpolation condition.

For regression models, parameter tuning is typically carried out by minimising an error measure calculated with respect to the deviation of the predicted surface from the true response. From a statistical point of view, the parameters can be set by maximising the probability that the response values of the observation data set are generated by the RSM, see [1].

In the RSM, $\hat{y} = y + \epsilon$, ϵ is the prediction error. With the assumption of independent and normally distributed errors ϵ_i , with zero mean and standard deviation σ , the probability P that the RSM yields $y_i + \epsilon_i$ as the response to the input \mathbf{x}_i can be written as,

$$P = \frac{1}{(2\pi\sigma^2)^{n/2}} \prod_{i=1}^n \left\{ \exp \left[-\frac{1}{2} \left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{\sigma} \right)^2 \right] \epsilon_i \right\}, \quad (12.6)$$

where n is the number of observations. Alternatively, the maximisation problem can be turned into a minimisation problem by calculating the negative natural logarithm of the above probability,

$$\min_{\alpha} \left[\sum_{i=1}^n \frac{(y_i - \hat{f}(\mathbf{x}_i))^2}{2\sigma^2} - n \ln \epsilon_i \right]. \quad (12.7)$$

12.2.3.4 Model Validation

Model validation is important for getting insight into the model fidelity. In the model training stage, the optimal model parameters are found. This stage, however, assesses the quality of the RSM by quantifying the error between the predicted and the true response on new data points that are different from the training points.

Some of the most used performance indices of RSM validation are introduced in Table 12.1.

Residuals are the differences between actual response values y_i and those predicted by the model on the data points. Residuals, as in the RSA and RSS measures, can be used to assess the quality of the fit. The smaller the RSA or RSS measures, the better the fit. TSS is the sum of the deviations of the observed responses from their sample average \bar{y} . The maximum value of R^2 is 1 in case of a perfectly interpolating model ($RSS = 0$). The closer the R^2 to the maximum value 1, the better the fit. However, the R^2 measure can be misleading. If the number of model parameters is too high compared to the number of observations available, R^2 increases to a value close to 1 because of overfitting. The surrogate model will provide good results only in the proximity of the observation points. It is therefore important to use metrics that penalise the number of degree of freedom p of the model, as in the case of the R^2_{adj} ,

Table 12.1 Performance indices

Performance index	Abbreviation	Definition
Residual	ϵ	$\epsilon_i = y_i - \hat{y}_i$
Residual sum of absolute error	RSA	$RSA = \sum y_i - \hat{y}_i $
Residual sum of squares	RSS	$RSS = \sum (y_i - \hat{y}_i)^2$
Total sum of squares	TSS	$TSS = \sum (y_i - \bar{y})^2$
Coefficient of determination	R^2	$R^2 = 1 - \frac{RSS}{TSS}$
Adjusted coefficient of determination	R_{adj}^2	$R_{adj}^2 = 1 - \frac{n-1}{n-p} \frac{RSS}{TSS}$
Akaike information criterion	AIC	$AIC = n \ln \frac{RSS}{n} + 2p$
Bayesian information criterion	BIC	$BIC = n \ln \frac{RSS}{n} + p \ln n$

AIC and BIC . Similarly to R^2 , the closer the R_{adj}^2 to 1 the better the fit, but this measure also has an $\frac{n-1}{n-p}$ term to penalise overfitting. AIC and BIC can assume negative values. Models with the smallest AIC or BIC values are preferable.

The above performance indices give a biased quality measure if they are calculated on the training data. Common practice is to divide the sample data into a training data set and a validation data set. Typically one quarter of the data is kept for validation. When separating this data, it requires great care to produce two, well-distributed data sets.

In the event of limited sampling data, the Leave-One-Out or Prediction Error Sum of Squares (PRESS) cross-validation techniques are recommended in that they use all sample points for training. These techniques systematically leave out sample instances or a group of samples and quantify the error between the trained response surface and the excluded observation data,

$$PRESS = \sum_i^n (y_i - \hat{y}_{(i)})^2, \tag{12.8}$$

where $\hat{y}_{(i)}$ is the response predicted by leaving out the i^{th} observation form the training set.

12.3 Examples of Response Surface Models

12.3.1 Least Squares Method

The Least Squares Method (LSM) minimises the RSS of RSM regression in that it minimises the sum of squared distances of the RSM and the true response evaluated in the training points.

Assuming the following form for the RSM function:

$$\hat{y} = \hat{f}(\mathbf{x}, \boldsymbol{\alpha}), \quad (12.9)$$

where $\hat{f}(\mathbf{x}, \boldsymbol{\alpha})$ is the RSM function, $\mathbf{x} = [x_1, \dots, x_m]$ is the vector of independent input variables, and $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_p]$ is the vector of model parameters to be adjusted, the LSM problem can be formulated as follows:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i, \boldsymbol{\alpha}))^2, \quad (12.10)$$

where n is the number of observations and y_i are the actual responses. The sum in Eq. (12.10) corresponds to the *RSS* index.

The parameters of the RSM can be determined according to the general minimum condition equation. The *RSS* index will reach its minimum where its first derivative equals zero,

$$\frac{\partial RSS}{\partial \alpha_j} = -2 \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i, \boldsymbol{\alpha})) \frac{\partial \hat{f}(\mathbf{x}_i, \boldsymbol{\alpha})}{\partial \alpha_j} = 0, \quad j = 1, \dots, p. \quad (12.11)$$

Further details on LSM can be found in Cavazzuti's book [3].

In case of a Linear Least Squares Problem (Linear LSM) the regression model can be written as,

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\alpha}, \quad (12.12)$$

where \mathbf{X} is a matrix of functions containing the basis functions that depend only on input variables. The model is linear because it is a linear combination of the basis functions: it is linear in the model coefficients, while the basis functions can be nonlinear.

By introducing the linear model of Eq. (12.12) in Eq. (12.10), the Linear Least Squares Problem is formulated as,

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^n (y_i - \mathbf{X}\boldsymbol{\alpha})^2. \quad (12.13)$$

According to Eq. (12.11) the model parameters can be obtained from the following equation:

$$\frac{\partial}{\partial \boldsymbol{\alpha}} (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha}) = -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X}\boldsymbol{\alpha} = 0. \quad (12.14)$$

The coefficients of the RSM can be given in closed form,

$$\alpha = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (12.15)$$

In the case of a Nonlinear Least Squares Problem the Eq. (12.11) is approximated and the coefficients are traditionally found by an iterative method such as the Gauss–Newton, Levenberg–Marquardt or Genetic Algorithms.

12.3.2 Radial Basis Functions

Radial Basis Function models are linear combinations of functions centred at the location of the training points,

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^n \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|/\delta), \quad (12.16)$$

where $\phi(\cdot)$ is the radial basis function which depends on the Euclidean distance from the centres \mathbf{x}_j and δ is a scaling parameter.

RBF models are typically used for interpolation, hence the coefficients of the model are determined by solving the interpolation equation,

$$\mathbf{f} = \mathbf{A}\alpha, \quad (12.17)$$

where $A_{ij} = \phi(\|x_i - x_j\|/\delta)$ are the entries of the collocation matrix, \mathbf{A} , representing the value of the j^{th} basis function evaluated in the location of the i^{th} point of the training dataset.

The collocation matrix must be positive definite to solve the above equation. Otherwise, when it is a conditionally positive definite collocation matrix, the radial basis functions are supplemented by an additional polynomial term,

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^n \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|/\delta) + p_m(\mathbf{x}), \quad (12.18)$$

$$p_m(\mathbf{x}) = \sum_{j=1}^q \beta_j \pi_j(\mathbf{x}), \quad q = \binom{m+d}{d}, \quad (12.19)$$

where m is the maximum polynomial degree and d is the number of variables and $\pi_j(\mathbf{x})$ are the polynomial basis functions and β_j are the coefficients of the polynomials. In the augmented RBF formulation, Eq. (12.19) is the so-called moment condition of the coefficients and it is necessary to determine the unknown coefficients in the interpolation equation.

Table 12.2 Typical basis functions for RBF models

Function type	Abbreviation	Definition
Gaussian	G	$\phi(r) = \exp^{-r^2}$
Thin plate splines	TPS	$\phi(r) = r^2 \log(r)$
General polyharmonic splines	PS	$\phi(r) = \begin{cases} r^k & \text{for } k \text{ odd} \\ r^k \log(r) & \text{for } k \text{ even} \end{cases}$
Multi-quadrics	MQ	$\phi(r) = (1 + r^2)^{1/2}$
Inverse multi-quadrics	IMQ	$\phi(r) = (1 + r^2)^{-1/2}$
Wendland compactly supported	W2	$\phi(r) = \begin{cases} (1 - r)_+^3(3r + 1) & \text{for } d = 1 \\ (1 - r)_+^4(4r + 1) & \text{for } d = 2, 3 \\ (1 - r)_+^5(5r + 1) & \text{for } d = 4, 5 \end{cases}$

The optimal model parameters are obtained by solving the augmented system of equations,

$$\begin{Bmatrix} \mathbf{f} \\ 0 \end{Bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix} \begin{Bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{Bmatrix}, \tag{12.20}$$

where $P_{ij} = \pi_j(\mathbf{x}_i)$ are the entries of \mathbf{P} , denoting the j^{th} polynomial basis function evaluated at \mathbf{x}_i and $\boldsymbol{\beta} = [\beta_1, \dots, \beta_q]$ is the vector of polynomial coefficients.

The complexity of RBF is ruled by the complexity of the matrix inversion and depends on the number of nodes, which tend to be proportional to the condition number of the matrix. Typically, RBF is characterised by a fast rate of convergence. For smooth RBFs the convergence rate is $O(\frac{-C_l}{h})$, where C_l is a constant and h is inversely proportional to n . Some typical basis functions for RBF models are given in Table 12.2, where $(\cdot)_+$ denotes an ordinary ramp function and k is an arbitrarily chosen integer parameter.

12.3.3 Kriging

The Kriging algorithm is commonly used to build interpolating response surfaces based on Gaussian Processes (GP). The Kriging algorithm was devised for Geostatistical purposes [7] but it has taken root in many other fields of science and engineering. Kriging provides a response surface prediction to model spatial variability. Since the prediction is characterised by stationarity, the estimated response depends only on the spatial distances of the predicted point and the training points. Therefore, Kriging is most accurate when in close proximity to the training points.

There are several versions of Kriging in literature. They are members of the GP regression model family, which is an extensively studied field. For further details

see the work of Rasmussen and Williams [8]. Forrester et al. book [1] discusses the methodology from an engineering point of view, whereas Cressie [9] puts the discussion into the original geostatistical context.

A GP regression model approximates the response as a stochastic process,

$$\hat{f}(\mathbf{x}) \sim GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (12.21)$$

where $GP(\cdot)$ means a Gaussian Process function, i.e. any finite linear combination of random variables with Gaussian joint probability density function. $\mu(\mathbf{x})$ denotes the trend or the mean and represents the global variations of the response, while $k(\mathbf{x}, \mathbf{x}')$ is the covariance function and represents the local variations of the response. In principle, any GP can be completely described by its second-order statistics, i.e. the mean and covariance functions. In addition to the global trend and local departure, the model can be extended by a white noise term in the event of noisy observations, in which case, the RSM can be formulated as follows:

$$\hat{f}(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}) + \epsilon(\mathbf{x}), \quad (12.22)$$

where $\mu(x)$ is the global trend, $\delta(x)$ is the local departure, and $\epsilon(x)$ is the white noise function, respectively. Both local departure and white noise terms have zero mean and constant variance.

In Ordinary Kriging, the mean function is constant and the local departure is stationary and depends only on the spatial difference of the samples. In practice, the white noise term is incorporated into the local departure term through an offset of the variogram, which is discussed in Sect. 12.3.3.1. Ordinary Kriging interpolates the response values as a linear combination of the observed values,

$$\hat{f}(\mathbf{x}) = \boldsymbol{\lambda}^T \mathbf{f}, \quad (12.23)$$

where $\boldsymbol{\lambda}$ are the weight parameters so that $\sum \lambda_i = 1$. The weights contain the term of local departure and as a result depend on the spatial difference of the training points and the predicted points. The optimal weights are computed by looking for the Best Linear Unbiased Estimator by minimising the Mean Squared Error of the predictor,

$$\min_{\boldsymbol{\lambda}} E[|\hat{f}(\mathbf{x}_*) - f(\mathbf{x}_*)|^2], \quad (12.24)$$

where \mathbf{x}_* is the location of the predicted value. The unbiasedness means that the approximation model and the true function evaluations have the same expected value (see Ref. [10]).

The general condition for the minimum yields the optimal weights [9],

$$\boldsymbol{\lambda}_{opt} = \mathbf{C}^{-1} \left[\mathbf{c} + \mathbf{1} \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{c}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \right], \quad (12.25)$$

where \mathbf{C} is the stationary covariance matrix calculated on the observed values having entries $C_{ij} = \text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for all $i, j = 1, \dots, n$. c is the vector of the stationary covariance between the predicted value and the observed points having entries $c_i = \text{cov}(\mathbf{x}_*, \mathbf{x}_i)$. They are determined by the variogram. $\mathbf{1}$ is the vector of ones.

The stationary covariance function is usually unknown and it is estimated by defining a variogram such that,

$$\text{cov}(\mathbf{x}_i, \mathbf{x}_j) = C(h) := \sigma - \gamma(h), \quad (12.26)$$

where $h = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the spatial distance of \mathbf{x}_i and \mathbf{x}_j . $\gamma(h)$ is the variogram and σ is the so-called *sill* which represents the maximum (global) variance of the observation data.

By substituting Eq. (12.25) into Eq. (12.23) the interpolation surface is

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{c}^T \mathbf{C}^{-1}(\mathbf{f} - \hat{\mu}\mathbf{1}), \quad (12.27)$$

where the mean is

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{f}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}, \quad (12.28)$$

and the squared standard deviation of the RSM is

$$\hat{\sigma}^2 = \frac{(\mathbf{f} - \hat{\mu}\mathbf{1})^T \mathbf{C}^{-1} (\mathbf{f} - \hat{\mu}\mathbf{1})}{n}. \quad (12.29)$$

12.3.3.1 Variogram

The variogram models the variance of random data at the given separation vector \mathbf{h} , as it was defined by [11].

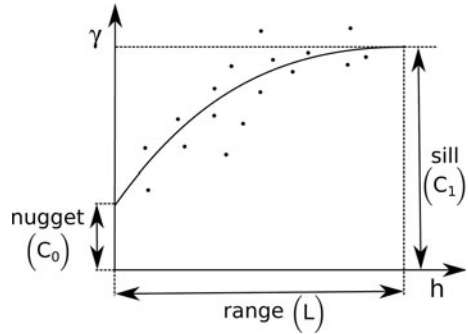
The empirical variance of a data set can be calculated:

$$s^2 = \frac{1}{2} \frac{1}{n(n-1)} \sum_{i \neq j} (y_j - y_i)^2. \quad (12.30)$$

The following experimental variogram is obtained by substituting $y_i = y(\mathbf{x}_i)$ and $y_j = y(\mathbf{x}_i + \mathbf{h})$ in the above equation:

$$\gamma(\mathbf{h}) = \frac{1}{2} \frac{1}{N(\mathbf{h})} \sum_{i=1}^{N(\mathbf{h})} (y_j - y_i)^2, \quad (12.31)$$

Fig. 12.3 Variogram



where $N(\mathbf{h})$ is the number of observed (y_i, y_j) pairs separated by \mathbf{h} vectors. In practice, isotropy is considered. The observations are grouped according to spatial distance categories whereas the variance of the observed data, at given separation distances, is calculated as follows:

$$\gamma(h) = \frac{1}{2} \frac{1}{N(h)} \sum_{i=1}^{N(h)} (y_j - y_i)^2, \tag{12.32}$$

where $h = \|\mathbf{h}\|$.

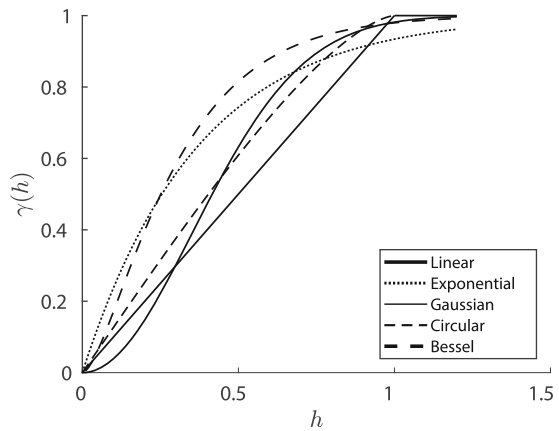
The experimental variogram $\gamma(h)$ can be approximated by various models; however, some common characteristics should be respected. By definition, the variogram gives zero if the separation distance is zero. The variogram is defined only for positive separation distance values. For values higher than zero, it increases monotonically from C_0 to $C_0 + C_1$. The lower bound C_0 (a.k.a. *nugget*) represents the white noise error term of Eq. (12.22). If $C_0 = 0$, no error is assumed in the observation data and Kriging interpolates the training data. On the other hand a high C_0 value results in a smooth response surface, even for noisy data. The upper bound is the sill represents the maximum (global) variance of the observation data. In other words, the variance of the response increases as the distance from the observation point increases until the global variance is reached. A general variogram curve can be seen on Fig. 12.3. The *range* is the separation distance measured when the sill is reached. If the sill is an asymptote of variogram model (as it is in exponential variograms), a practical range can be defined as the distance h for which $\gamma(h)$ is 95% of the sill. The range is inversely proportional to the problem complexity. Small ranges result in sudden response variations, while large ranges result in slow response variations. In special cases, the characteristics of the variogram can differ from the above described properties.

For commonly used variogram models see Table 12.3 and Fig. 12.4, where L is the range and K_1 is a first order Bessel function of the second kind.

Table 12.3 Typical variogram models

Variogram model	Definition
Linear	$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ C_0 + C_1 \left(\frac{h}{L}\right) & \text{for } 0 < h < L \\ C_0 + C_1 & \text{for } h \geq L \end{cases}$
Exponential	$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ C_0 + C_1 \left(1 - e^{-\frac{h}{L}}\right) & \text{for } h > 0 \end{cases}$
Gaussian	$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ C_0 + C_1 \left(1 - e^{-\frac{h^2}{L^2}}\right) & \text{for } h > 0 \end{cases}$
Circular	$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ C_0 + C_1 \left(\frac{2}{\pi} \frac{h}{L} \sqrt{1 - \left(\frac{h}{L}\right)^2} + \frac{2}{\pi} \arcsin\left(\frac{h}{L}\right)\right) & \text{for } 0 < h < L \\ C_0 + C_1 & \text{for } h \geq L \end{cases}$
Bessel	$\gamma(h) = \begin{cases} 0 & \text{for } h = 0 \\ C_0 + C_1 \left(1 - \frac{h}{L} K_1\left(\frac{h}{L}\right)\right) & \text{for } h > 0 \end{cases}$

Fig. 12.4 Typical variograms (scaled to have range $L = 1$)



12.4 Wing Structure Design Using Response Surface Models

12.4.1 Design Problem

A radically simplified wing design optimisation problem is introduced in this section to show the use of RSMs in optimisation.

The chosen optimisation problem aims at finding the optimal angle-of-attack (α) which maximise the ratio of the lift C_L and drag C_D coefficients of a Beechcraft Baron 58 aircraft [12]. The optimisation problem is formulated as:

$$\begin{aligned} \max_{\alpha} \quad & \frac{C_L(\alpha)}{C_D(\alpha)} \\ \text{s.t.} \quad & 0 \leq \alpha \leq 10 [^\circ]. \end{aligned} \quad (12.33)$$

The angle-of-attack can be defined using various reference lines. Here, it is defined as the angle between the velocity vector of the aircraft and the chord line of the airfoil of the wing section. Further, it is assumed that the wing has the same airfoil section, chord length and twist through the entire wingspan. The twist of the wing is optimised for level flight when the longitudinal axis of fuselage of the aircraft is horizontal, see Fig. 12.5.

12.4.2 Analytical Model

The lift coefficient of a finite 3-D wing is calculated according to the following equation:

$$C_L = a(\alpha - \alpha_{L=0}), \quad (12.34)$$

where $\alpha_{L=0}$ is the angle-of-attack resulting in zero-lift and the slope a of the lift curve of the finite 3-D wing is obtained according to the Helmbold formula [12]:

$$a = \frac{a_0}{\sqrt{1 + \left(\frac{a_0}{\pi AR}\right)^2 + \frac{a_0}{\pi AR}}}, \quad (12.35)$$

where a_0 is the slope of the lift curve of the 2-D airfoil section and considered to be equal to its theoretical value 2π . AR denotes the aspect ratio of the wing (i.e. the

Fig. 12.5 The wing design problem (left side view)

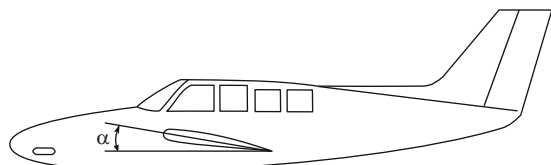
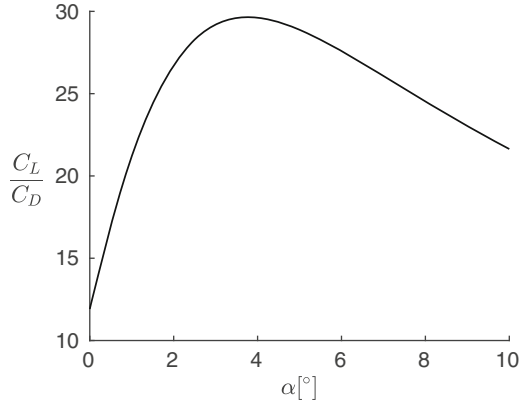


Fig. 12.6 Analytical curve

ratio of the square of the wingspan divided by the wing area). The drag coefficient is calculated as a sum of the zero-lift drag coefficient C_d and the induced drag,

$$C_D = C_d + \frac{C_L^2}{\pi AR}. \quad (12.36)$$

Therefore, the objective of the optimisation problem is

$$\frac{C_L}{C_D} = \frac{a(\alpha - \alpha_{L=0})}{C_d + \frac{a^2(\alpha - \alpha_{L=0})^2}{\pi AR}}, \quad (12.37)$$

which is plotted in Fig. 12.6.

The optimal angle-of-attack of the Beechcraft Baron 58 aircraft is

$$\alpha_{opt} = 0.0658 [rad] = 3.768 [^\circ], \quad (12.38)$$

and provides a lift to drag ratio $\frac{C_L}{C_D} = 29.647$ by analytically calculating the optimum from Eq. (12.37), where $\alpha_{L=0} = -0.0175 [rad]$, $C_d = 0.0068 [-]$ and $a_0 = 2\pi \left[\frac{1}{rad} \right]$.

12.4.3 Comparison of Response Surface Models

Suppose that the analytical model is not available but by measurement for any α the corresponding lift to drag ratio can be calculated. The resources are limited such that only $n = 25$ observations can be conducted. Therefore, 25 randomly sampled α values were generated and the corresponding lift to drag ratio was acquired. With the available observation points three RSMs, namely Linear LSM, Kriging, and RBF, were built to make an approximation of the entire design space, see Fig. 12.7b, c, d.

The trainings of RSMs were performed using the multi-disciplinary optimisation software tool modeFRONTIER with default parameters [13].

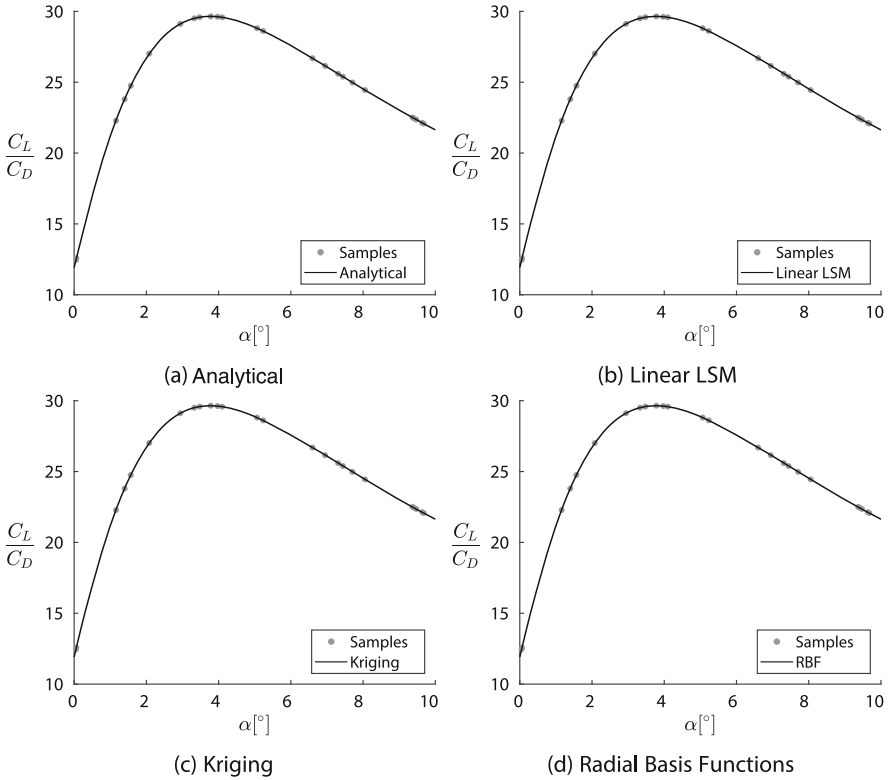


Fig. 12.7 Lift to drag ratio ($\frac{C_L}{C_D}$) as a function of the angle-of-attack (α): analytical model response (a) and RSMs (b)–(d)

According to Fig. 12.7, there is no significant difference between the RSMs. This fact is supported by the quality indicators shown in Table 12.4, where an additional database of 10 Uniform Latin Hypercube Sampling (ULHS) samples was used for validation. All the RSMs have an R^2 value close to 1 given the simplicity of the investigated problem. Therefore, in this case the R^2 value does not provide enough information about the quality of the RSMs to differentiate between them.

RBF has the lowest AIC value and thus it provides the most accurate approximation; however, the Linear LSM and Kriging have AIC value of the same order of magnitude. The Linear LSM is evaluated slightly better than Kriging as Linear LSM uses less parameters to achieve a comparable error magnitude.

In the investigated RSMs not only the number of the parameters but also the choice of the basis functions is an important factor for the quality of the approximation. For the case of Linear LSM, the performance of different polynomial orders was studied. Results are summarised in Table 12.5, where it is shown that higher polynomial orders result in better approximations. The 10th order was selected for the benchmark of Table 12.4. Analogously, RBF and Kriging can produce different

Table 12.4 Performance indices of RSMs of lift to drag ratio by using a validation set of 10 samples generated with ULHS, where $\frac{RSA}{n}$ is the mean absolute error (averaged over n validation points)

RSM	$\frac{RSA}{n}$	R^2	AIC
Linear LSM (10th order)	1.07E-4	1.000	-1.54E2
Ordinary Kriging (Gaussian)	2.61E-5	1.000	-1.41E2
RBF (Hardy MQ)	3.55E-7	1.000	-2.34E2

Table 12.5 Comparison of various basis functions of Linear LSM by using a validation set of 10 samples generated with ULHS, where $\frac{RSA}{n}$ is the mean absolute error

Linear LSM	$\frac{RSA}{n}$	R^2	AIC
1st order	3.92	5.08E-2	3.57E1
2nd order	1.67	8.62E-1	1.84E1
5th order	3.04E-2	1.000	-5.13E1
10th order	1.07E-4	1.000	-1.54E2

Table 12.6 Optimal designs obtained with different RSMs and the analytical response. A column with the relative errors calculated with respect to the analytical optimal results is provided for the angle-of-attack (third column) and lift to drag ratio (fifth column)

Model	α [°]	Rel. error	$\frac{C_L}{C_D}$	Rel. error
Analytical	3.767906	-	29.647141	-
Linear LSM (10th order)	3.764428	-0.092%	29.647104	-1.25E-4%
Ordinary Kriging	3.773268	0.142%	29.647122	-6.41E-5%
RBF	3.766315	-0.042%	29.647139	-6.75E-6%

outputs with different basis functions or variograms; however, in this simple case the differences are not significant. The default basis functions of modeFRONTIER were applied.

The RSMs found above were used as a black-box for design evaluations to optimise the problem (12.33). Three optimisation runs were performed, one for each RSMs. The analytical optimum α_{opt} of Eq. (12.38) was used as a reference. All the optimisations were performed with the NSGA-II algorithm [14] available in modeFRONTIER with default parameters, 50 generations, and the same initial population of 10 individuals generated with a ULHS algorithm.

Table 12.6 shows the difference in the optimal designs found. Interestingly, Kriging provides a better approximation (compared to Linear LSM) for the lift to drag ratio; however, the optimal angle-of-attack is better approximated by the Linear LSM. In our case we are more interested in the optimal angle-of-attack. Therefore, Linear LSM is preferred over Kriging in our case. The optimal value prediction performed with RBF is superior to Linear LSM and Kriging. This agrees with the provided RSM ranking by the AIC values.

Table 12.7 Performance indices of different RBF models of the lift to drag ratio. The validation set consists of 10 samples generated with ULHS

RSM	$\frac{RSA}{n}$	R^2	AIC
RBF-25 (random)	3.55E-7	1.000	-2.34E2
RBF-50 (random)	5.47E-8	1.000	-2.16E2
RBF-100 (random)	9.05E-9	1.000	-1.63E2

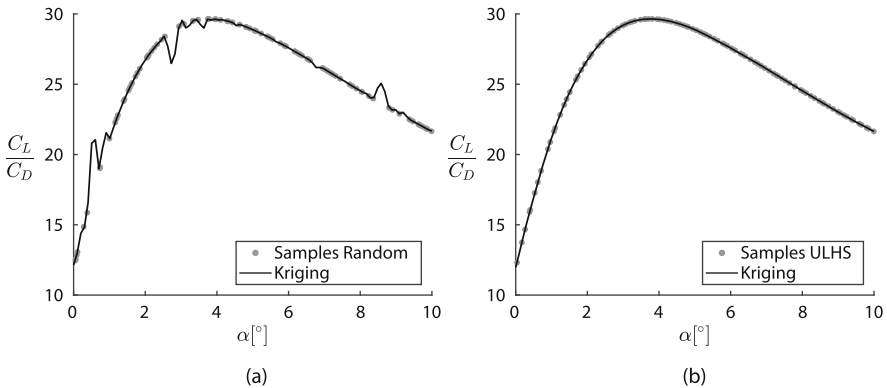


Fig. 12.8 Comparison of random sampling (a) and ULHS (b) for Kriging (100 samples)

If the available resource is increased to acquire more samples, it is expected to have a more accurate RSM. To check this expectation, two additional data sets were generated containing, respectively, 50 and 100 randomly sampled observation points. These data sets were used to construct two RBF models, called RBF-50 and RBF-100 in the following, while the RBF model previously constructed on the 25 data point set is named RBF-25. The performance indicators of the RBF models are presented in Table 12.7. As expected the mean absolute error decreases as n increases. The number of parameters in the RBF model increases with n , because the size of the collocation matrix increases proportionally to that number. Therefore, the AIC indicator deteriorates.

In the previous example, the sample data were generated randomly which provided a sufficiently good data source for the RBF. However, there are RSMs which are more sensitive to the distribution of the sample data. For example, constructing a Kriging RSM by using the 100 randomly sampled data points results in a distorted curve shown in Fig. 12.8a. The reason for this is that the randomly sampled data points are not well distributed. The distance between some points are very small which cause an ill-conditioned covariance matrix. This problem can be solved by generating the samples with ULHS which provides a well-distributed sample data and thus a better data source for Kriging, as shown in Fig. 12.8b. Alternatively, Universal Kriging can be used which is a regression method and would result in a smoother approximation as well [15].

12.4.4 RSM Construction on Noisy Data

In the previous section, the problem was considered uncertainty-free. In this section the applicability of RSM to noisy data is investigated. RSMs on noisy response data have to be treated carefully and the devoted literature should be studied [16]. Here, RSM training on stochastic data is not dealt with in its full deepness; only a limited sense of this wide topic is presented for the reader.

To mimic a stochastic problem, an additional random term was assumed stemming from various sources of uncertainty:

$$\frac{C_L}{C_D} = \frac{a(\alpha - \alpha_{L=0})}{C_d + \frac{a^2(\alpha - \alpha_{L=0})^2}{\pi AR}} + \epsilon, \tag{12.39}$$

where ϵ takes values between -0.5 and 0.5 randomly.

Here, analogously to previous case, the lift to drag ratio is the only measured value. Similarly, the resources are limited and 100 data points are available for

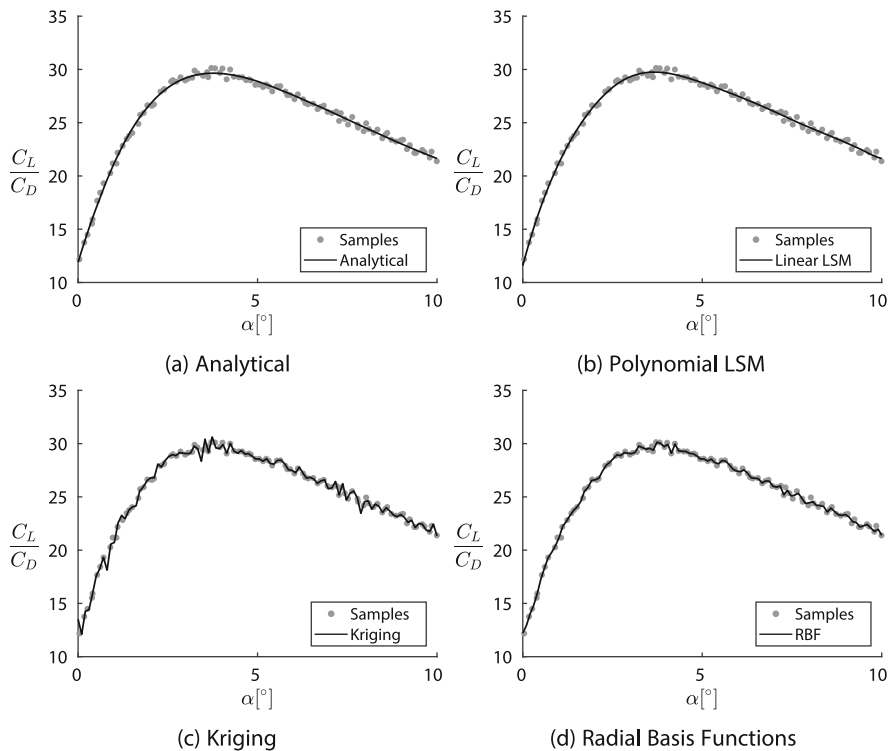


Fig. 12.9 Lift to drag ratio ($\frac{C_L}{C_D}$) as a function of the angle-of-attack (α): analytical model response (a) and RSMs (b)–(d)

Table 12.8 Performance indices of RSMs built on noisy data (the validation set of 25 samples were drawn with ULHS from the analytical) function

RSM	$\frac{RSA}{n}$	R^2	AIC
Linear LSM (10th order)	7.53E-2	9.99	-9.88E1
Ordinary Kriging (Gaussian)	3.75E-1	9.86	1.67E2
RBF (Hardy MQ)	2.38E-1	9.95	1.37E2

Table 12.9 Optimal designs obtained with different RSMs built on noisy data and the optimal analytical response value

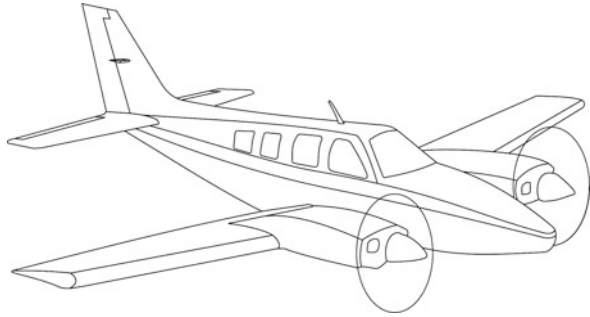
Model	α [°]	Rel. error	$\frac{C_L}{C_D}$	Rel. error
Analytical	3.767906	–	29.647141	–
Linear LSM	3.663040	-2.78%	29.755548	0.37%
Ordinary Kriging	3.760577	-0.19%	30.804586	3.90%
RBF	3.414064	-9.39%	30.135911	1.65%

RSM training. In this case, the validation set size is 25 and drawn from the analytical function with ULHS. In reality, the validation set is usually also loaded with uncertainty but for better comparison the analytical function is used here for validation. The 100 observation points were generated with ULHS and evaluated with Eq. 12.39. The results form a well-distributed noisy data-cloud around the analytical curve as it can be seen in Fig. 12.9a. The RSMs generated from this noisy data are obviously less accurate than their uncertainty-free counterparts. From Table 12.8 we can see that all three RSMs have performance indices of the same magnitude. The only exception is the AIC value of the Linear LSM which indicates a better approximation than the other two RSMs as it has a significantly smaller number of model parameters. The decreased accuracy of the RSMs can be also observed in the obtained optimal values as shown in Table 12.9. The optimisation runs were performed with the same criteria as in Sect. 12.4.3. Following the RSM ranking of the AIC values Linear LSM provides the best approximation for the lift to drag ratio and Kriging is slightly worse than RBF. The obtained RSMs are plotted in Fig. 12.9b, c, d. We can see that Kriging and RBF provide oscillatory curves as they are interpolators of the data. Typically, interpolation RSM is not suggested on raw noisy data. The noisy data should be filtered and the filtered data can be approximated by an interpolation method. Regression, however, can provide a viable option even on noisy data as it inherently filters the noise [1].

12.4.5 Case-Study Conclusion and Take-Home Message

The analytical optimal value for the angle-of-attack of the wing of the Beechcraft Baron 58 (Fig. 12.10) could not be obtained by any RSMs without error. The RSM based optimal values, however, provided good approximations of the true optimal

Fig. 12.10 Beechcraft Baron 58 (illustration)



value. In case of an expensive experiment or numerical analysis, RSM techniques are highly valuable mathematical tools. For a small decrement in the accuracy, RSMs can save great amount of time solving a complex problem.

The best approximation of the uncertainty-free problem showed 0.042% error using RBF while the Kriging performed the worst. By considering uncertainty, the Linear LSM proved to be a slightly better tool to approximate the optimal angle-of-attack. In case of uncertainty, the interpolation methods, Kriging and RBF, resulted in an oscillatory curve which is typically avoidable. Therefore, in case of uncertainty, regression methods are favoured or an additional data filtering stage is required.

In real-world applications, the problems are typically more complex and therefore the quality of the RSM is more difficult to assess. Performance indices like R^2 or AIC help to improve our understanding of the accuracy and quality of RSMs; however, deeper analysis is often required, particularly in a problem affected by uncertainty. RSMs are important techniques in many fields. The proper choice of which RSM algorithm to use is non-trivial and depends on the investigated problem.

References

1. A. Forrester, A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide* (Wiley, New York, 2008)
2. R.M. Paiva et al., Comparison of surrogate models in a multidisciplinary optimization framework for wing design. *AIAA J.* **48**(5), 995–1006 (2010)
3. M. Cavazzuti, *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics* (Springer Science & Business Media, Berlin Heidelberg, 2012)
4. R.H. Myers, D.C. Montgomery, C.M. Anderson-Cook, , A.: *Response Surface Methodology: Process and Product Optimization Using Designed Experiments* (Wiley, Hoboken, NJ, 2016)
5. A. Saltelli, K. Chan, E.M. Scott et al., *Sensitivity Analysis* (Wiley, New York, 2000)
6. L. Eriksson, E. Johansson et al., *Design of Experiments. Principles and Applications* (Learn Ways AB, Stockholm, 2000)
7. D.G. Krige, A statistical approach to some basic mine valuation problems on the Witwatersrand. *J. South. Afr. Inst. Min. Metall.* **52**(9), 201–203 (1951)
8. C.K. Williams, C.E. Rasmussen, Gaussian processes for regression, in *Advances in Neural Information Processing Systems* (1996)

9. N. Cressie, *Statistics for Spatial Data* (Wiley, New York, 2015)
10. S. Puntanen, G. Styan, The equality of the ordinary least squares estimator and the best linear unbiased estimator. *Am. Stat.* **43**, 151–161 (1989)
11. M. Bachmaier, M. Backes, Variogram or semivariogram? Understanding the variances in a variogram. *Prec. Agric.* **9**(3), 173–175 (2008)
12. J. D. Anderson Jr, *Fundamentals of Aerodynamics* (Tata McGraw-Hill Education, New York, 2010)
13. modeFRONTIER 2018R1, Esteco SpA. <http://www.esteco.com/modelfrontier>
14. K. Deb et al., A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002)
15. R. Christensen, *Linear Models for Multivariate, Time Series, and Spatial Data* (Springer Science and Business Media, New York, 1991)
16. A.I. Forrester, A.J. Keane, Recent advances in surrogate-based optimization. *Progr. Aerosp. Sci.* **45**(1–3), 50–79 (2009)

Chapter 13

Risk Measures in the Context of Robust and Reliability Based Optimization



Elisa Morales Tirado and Domenico Quagliarella

Abstract The base concepts of robust optimization are described and detailed in this chapter. In particular, the approach based on risk measurements is introduced, after a first quick review of the classical deterministic approach. In this context, the use of some special classes of risk measures, used in financial engineering, is reported along with the main advantages and drawbacks related to their mathematical features. The usage of these risk measures will be then illustrated in an example problem of robust aerodynamic design optimization. The focus is also given to advanced techniques for error and confidence interval estimations and how they can be used in the context of robust optimization to improve the overall efficiency and effectiveness of the process.

Keywords Risk measure · Robust optimization · Aerodynamic shape design · Coherent risk measure

13.1 Introduction

Solving optimization problems in the presence of uncertainty is of fundamental importance in many research and application fields ranging from economics [20] to engineering [14] to decision theory. Uncertainty can be present at various levels in an optimization problem and influence its solution in different ways. An important and significant source of uncertainty can be, to begin with, intrinsic to the mathematical model adopted to describe the physical [7], biological or economic system on whose parameters one wants to intervene to alter its behaviour by adapting it to their needs. This one is a case of epistemic uncertainty, which has distinct characters respect to stochastic uncertainty. The latter originates in the

E. M. Tirado (✉) · D. Quagliarella
Italian Aerospace Research Centre, Capua, Italy
e-mail: e.moralestirado@cira.it; d.quagliarella@cira.it

indeterminacy invariably present in the initial and operating conditions [21] of the system to be controlled.

Furthermore, in an industrial product, there are often manufacturing tolerances or deviations from the nominal characteristics of the components that are not quickly and economically eliminable. Therefore, an optimization process that allows taking into account the sources of uncertainty and their effects on the performance to be improved is often vital for many fields of human activities, from logistics to economics to engineering and medicine. There are many approaches to robust optimization, and some are well established. However, the focus of scientific literature is often on improving the computational performance of the optimization processes in conditions of uncertainty [10, 19]. Indeed, in this type of procedure, the quantity of interest is a statistical measure that requires the evaluation of a sample of the population elements, often quite large to reach acceptable reliability.

Here, instead, rather than on efficient sampling processes, the focus is on the robust optimization problem definition itself. In particular, the primary theme is the use of the so-called risk functions and their choice according to the robust optimization problem at hand. In particular, the use of the Value-at-Risk (VaR) and the Conditional Value-at-Risk (CVaR) is introduced. These risk functions were introduced and are still widely used in financial engineering. However, as this chapter illustrates, they are well suited to describe and formalize optimization problems in different engineering sectors [13]. In particular, their versatility is here demonstrated by applying CVaR to the performance improvement of a wing section.

13.2 Optimization Under Uncertainty

Optimization problems are mathematically defined using the minimization formulation referred below:

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{z}) \\ \mathbf{z} \in S \\ \text{s.t.} : \\ \quad c_i(\mathbf{z}) \leq 0 \quad i = 1, \dots, m \\ \quad S \subseteq \mathbb{R}^n \end{array} \right. \quad (13.1)$$

where the objective function is $f(\mathbf{z})$ and the vector \mathbf{z} is referred to as the vector of design variables. Furthermore, the objective function may be subject to constraint functions, expressed by $c_i(\mathbf{z})$. Constraints functions can be linear or nonlinear functions of the design variables, and these functions can be either explicit or implicit in \mathbf{z} . A quite common convention, which does not affect the generality of the formulation, is to represent all the inequalities as non-positive ones. In addition, the problem (Equation 13.1) has been presented as a minimization problem, but

some optimization problems might require maximization. Indeed, the maximization of $f(\mathbf{z})$ is always equivalent to the minimization of $-f(\mathbf{z})$ [18].

However, in most of the engineering problems, unknowns or future states must be considered. Moreover, they must account for the stochastic nature of the system and processes to be designed. For example, industrial manufacturing processes and real operating conditions inevitably introduce tolerances in the production and uncertainties in the working conditions, respectively, that will lead to deviations from the considerations taken at design stage. Hence, random variables are introduced, and a stochastic optimization problem is defined to correctly model the process under investigation.

A random variable is defined as a measurable function $X : \Omega \mapsto \mathbb{R}$ that maps possible outcomes Ω to a measurable space \mathbb{R} , with (Ω, \mathcal{F}, P) a properly defined probability space, with $\omega \in \Omega$, $\mathcal{F} = 2^\Omega$, and P a probability measure [4].

Mathematically, the direct introduction of random variables into the optimization problem (Eq. 13.1) introduces a functional dependency that has to be properly treated to avoid inconsistencies. Indeed, the introduction of random variables leads to the following problem formulation:

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{z}, X) \\ \mathbf{z} \in S \\ \text{s.to :} \\ \quad c_i(\mathbf{z}, X) \leq 0 \quad i = 1, \dots, m \\ \quad S \subseteq \mathbb{R}^n \end{array} \right. \quad (13.2)$$

where the objective function and the constraints are now functionals. Therefore, a way to recast the problem into an optimization one must be searched. Herein, several approaches are shown.

Best Estimate A particular outcome is chosen $\bar{\omega} \in \Omega$ as the best estimate of the unknown status. As a consequence, the problem is reconstructed as a deterministic optimization:

$$\left\{ \begin{array}{l} \min \quad f(\mathbf{z}, X(\bar{\omega})) \\ \mathbf{z} \in S \\ \text{s.to :} \\ \quad c_i(\mathbf{z}, X(\bar{\omega})) \leq 0 \quad i = 1, \dots, m \\ \quad S \subseteq \mathbb{R}^n \end{array} \right. \quad (13.3)$$

Although attractive for its simplicity, this kind of alternative is very risky, as the choice of the typical outcome $\bar{\omega}$ is somewhat arbitrary and might not reflect at all what happens in the reality.

Worst Case Contrary to the best estimate strategy, the worst possible outcomes are identified for the unknown status. This leads to the following minmax problem:

$$\left\{ \begin{array}{ll} \min & \sup f(\mathbf{z}, X(\omega)) \\ \mathbf{z} \in S & \omega \in \Omega \\ \text{s.to :} & \\ & c_i(\mathbf{z}, X(\omega)) \leq 0 \quad i = 1, \dots, m \\ & S \subseteq \mathbb{R}^n \end{array} \right. \quad (13.4)$$

Two main disadvantages of this method are that minmax problems are generally very computationally expensive and that the obtained solution is too conservative. In addition, there is a high probability to face a nonfeasible problem.

Expected Values with Safety Margins In this case, expectations, as well as standard deviations are introduced in a weighted sum. Therefore, the robust design problem is cast in the following form:

$$\left\{ \begin{array}{ll} \min & E[f(\mathbf{z}, X)] + \lambda_0 \sigma(f(\mathbf{z}, X)) \\ \mathbf{z} \in S & \\ \text{s.to :} & \\ & E[c_i(\mathbf{z}, X)] + \lambda_i \sigma(c_i(\mathbf{z}, X)) \leq 0 \quad i = 1, \dots, m \\ & S \subseteq \mathbb{R}^n \end{array} \right. \quad (13.5)$$

In this formulation, weighted sums of standard deviations can be interpreted as safety margins. Furthermore, this approach is widely used, although it can lead to serious problems such as the convergence to sub-optimal solutions due to the use of expectations that penalize favourable and unfavourable candidate solutions in the same way.

Performance Index on Cumulative Distribution Function (CDF) This approach is based on the definition of an ad hoc performance index (or risk measure) as a function of the Cumulative Distribution Function related to the quantity of interest under investigation. For the sake of completeness, let us give the definition of the Cumulative Distribution Function (CDF): the CDF gives the area under the probability density function from minus infinity to x . It describes the probability that a real-valued random variable X with a given distribution will be found at a value less than or equal to x . Mathematically, this is expressed by Eq. 13.6.

$$F_X(x) = P(X \leq x) \quad (13.6)$$

A performance index (or risk measurement) allows the comparison of different CDF shapes according to the risk criterion defined by the user. In this chapter, the risk measures used are the Value-at-Risk (VaR) and the Conditional Value-at-Risk (CVaR), also known as quantile and superquantile, respectively (their definitions are given in Sect. 13.3).

13.3 Risk Measures

When random events are modeled by random variables, as in the context of optimization under uncertainty, a way to measure risk should be figured out. With this purpose, a functional $\rho(X)$ is going to be defined for risk level quantification. Subsequently, an acceptable level of risk C must be decided, considering that there will, inevitably, be adverse events. Thus, the next inequality equation can be defined:

$$\rho(X) \leq C \quad (13.7)$$

Then, if the random variables representative of the cost depend on a deterministic decision vector \mathbf{x} of size m , the following minimization problem can be stated:

$$\left\{ \begin{array}{l} \min \quad \rho_0(X_0(x)) \\ \mathbf{x} \in S \subseteq \mathbb{R}^n \\ \text{s.t.} : \\ \rho_i(X_i(x)) \leq c_i \quad i = 1, \dots, m \end{array} \right. \quad (13.8)$$

Within this framework, different definitions for risk functional can be established. This will lead to different approaches of facing optimization problems under uncertainty. The most immediate and familiar alternative of risk measure is the expected value. This means that, on average, it should be $X \leq C$:

$$\mu(X) \leq C \rightarrow \rho(X) = \mu(X) = EX \quad (13.9)$$

Being more stringent, a condition on the standard deviation or on variance could be imposed if there is a need to reduce the variation range of the quantity of interest:

$$\mu(X) + \lambda\sigma(X) \leq C \rightarrow \rho(X) = \mu(X) + \lambda\sigma(X) \quad (13.10)$$

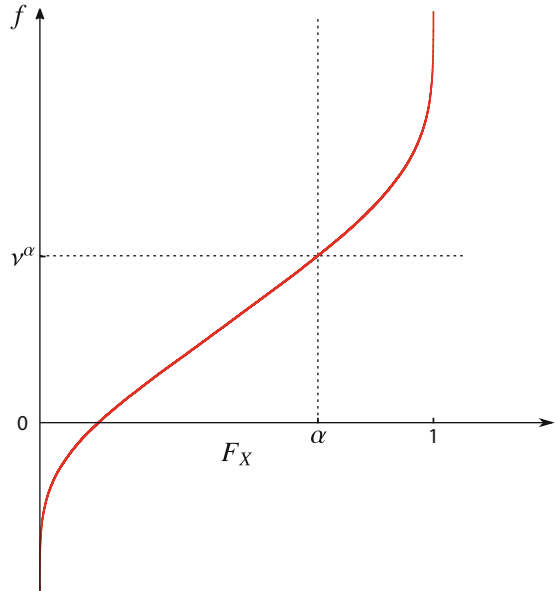
$$\mu(X) + \lambda\sigma^2(X) \leq C \rightarrow \rho(X) = \mu(X) + \lambda\sigma^2(X) \quad (13.11)$$

Indeed, classical robust design problem formulation is based on mean, μ , and variance, σ^2 [11], which can be treated as separated objectives in a multi-objective framework [9], as a weighted combination, or even cast into a constrained optimization format. However, the use of this classical approach may often generate some problems, since mean and variance are not independent measures, and it might be difficult to decide how much the mean must be penalized to get the desired reduction of variance.

Alternative risk measures are available that offer a better control on the desired features of the cumulative distribution function of interest. Here, in particular, the Value-at-Risk or the Conditional Value-at-Risk are introduced and used.

Let X be a random variable and $F_X(x) = P(X \leq x)$ the Cumulative Distribution Function of X . Thus, the inverse CDF of X can be defined as $F_X^{-1}(\gamma) = \inf\{x :$

Fig. 13.1 Value-at-Risk



$F_X(x) \geq \gamma$. This function gives the minimum value of x that makes the CDF of X to be greater than or equal to γ . Hence, α -VaR, i.e. the Value-at-Risk for a given $\alpha \in (0, 1)$, is given by

$$v^\alpha = F_X^{-1}(\alpha) \tag{13.12}$$

In other words, VaR is the maximum loss that can be exceeded only in a $(1 - \alpha)100\%$ of cases. In its definition, the infimum is used since CDFs are, usually, weakly monotonic and right-continuous. The α -VaR is shown in Fig. 13.1.

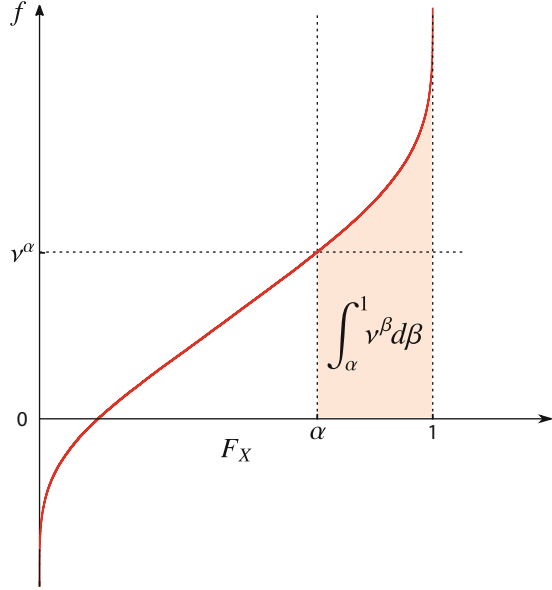
The definition of Conditional Value-at-Risk is given below. Let X be a random variable, the α -CVaR of X can be thought of as the conditional expectation of losses that exceed q_α . From a mathematical point of view, CVaR is given by a weighted average between α -VaR and the losses exceeding it. The comparison of VaR and CVaR shows that the latter is more sensitive to the shape of the upper tail of the cumulative distribution. Summing up, the CVaR is expressed as:

$$c^\alpha = \frac{1}{1 - \alpha} \int_\alpha^1 v^\beta d\beta \tag{13.13}$$

The area measured by the integral of the α -CVaR formula is highlighted in Fig. 13.2.

CVaR has the advantage, with respect to VaR, of being a coherent risk measure. The definition of coherency for a risk measure is a rigorous and well-defined mathematical concept that the interested reader can find in [2].

Fig. 13.2 Conditional Value-at-Risk



The properties that a risk measure must fulfil for being coherent [2] are enumerated below.

1. $\rho(C) = C$ for all constants C
2. *Convexity*: $\rho(1 - \lambda)X + \lambda' \leq (1 - \lambda)\rho(X) + \lambda\rho(X')$ for $\lambda \in (0, 1)$
3. *Monotonicity*: $\rho(X) \leq \rho(X')$ if $X \leq X'$.
4. *Closedness*: $\rho \leq c$ when $X_k \rightarrow X$ with $\rho(X_k) \leq c$
5. *Positive homogeneity*: $\rho(\lambda X) = \lambda\rho(X)$ for $\lambda > 0$.

From this last condition, the subsequent properties are derived:

- *Translation invariance*: $\rho(X + C) = \rho(X) + C$.
- *Sub-additivity*: $\rho(X + X') \leq \rho(X) + \rho(X')$.

Coherency properties offer several advantages in a robust optimization problem, and an actual robust aerodynamic shape design problem is here used to illustrate their meaning. The problem regards the improvement of the drag performance of a natural laminar flow wing and is described in detail in [12]. *Monotonicity* means that if the laminar performance of a generic wing X_2 is always better than X_1 , then the risk of X_2 is always inferior to the risk of X_1 . *Translation invariance* condition implies that a global delay of laminar to turbulent transition reduces the risk. Moreover, for a natural laminar flow wing-body, *sub-additivity* means that having two independent sources of laminarity (upper and lower wing surfaces) can only decrease risk. Summing up, coherency offers a mean to take into account the effect of desirable physical features in the risk measure used to formalize the robust design optimization problem to be solved.

Conversely, the Value-at-Risk is not a coherent measure since it does not respect the sub-additivity property.

13.4 Robust Optimization Problem Using Risk Functions

Risk measures, likewise expectations and variance, are unknown parameters of a statistical model (estimands), which can be only approximated using estimators and finite samples of data. Consequently, the robust optimization problem has to be defined in terms of estimates of the risk functions within the framework of multi-objective optimization. Therefore, Eq. 13.8 becomes

$$\left\{ \begin{array}{l} \min \quad \hat{\rho}_{i;n}(\mathbf{z}) \quad i = 1, \dots, p \\ \mathbf{z} \in Z \subseteq \mathbb{R}^n \\ s.t.o : \\ \hat{\rho}_{i;n}(\mathbf{z}) \leq c_i \quad i = p + 1, \dots, p + q \end{array} \right. \quad (13.14)$$

where $\hat{\rho}_{i;n}$ is an estimate of the generic risk measure ρ_i obtained using a sample of size n and a proper estimator. Moreover, the constraints are also given taking into consideration a set of inequalities which are defined in terms of q further risk measure estimates.

In addition, it must be mentioned that the quality of the risk function estimate will directly influence the results of the optimization problem. Hence, some guidelines should be followed when formulating a robust optimization problem [12]:

- when estimating risk functions, use the largest possible number of samples considering the computational budget;
- use advanced techniques for sampling (i.e. multilevel Monte Carlo or Control Variates);
- a low accuracy of the estimate can be perceived by the optimization algorithm as noise, thus select an optimization algorithm the least sensitive to noise as possible;
- employ advanced statistical methods for the evaluation of the estimate accuracy and confidence intervals. In particular, the bootstrap method will be adopted here (see Sect. 13.4.3).

The estimates of risk functions are here made either using the Empirical Cumulative Distribution Function (ECDF) or the Weighted Empirical Cumulative Distribution Function (WECDF), and the bootstrap is used to obtain accuracy and confidence intervals.

13.4.1 Estimation of Risk Functions Using ECDF

The Empirical Cumulative Distribution Function (ECDF) is the distribution function associated with the empirical measure of a sample. Moreover, it can be seen as a step function that jumps up by $1/n$ at each of the n data points. It takes as value the fraction of observations of the variable that are less or equal to the specified value [16].

Mathematically speaking, let $X : \Omega \mapsto \mathbb{R}^d$ a random variable, $\mathbf{x}^i = (x_1^i, \dots, x_d^i)$ a random sample of X , μ a probability measure, and $\mathbf{t} = (t_1, \dots, t_d)$ a generic vector in \mathbb{R}^d . The ECDF is defined in Eq. 13.15 for n samples $\{\mathbf{x}^1, \dots, \mathbf{x}^n\}$.

$$\hat{F}_\mu^n = \frac{\text{number of elements in the sample} \leq \mathbf{t}}{n} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{\mathbf{x}^i \leq \mathbf{t}\} \quad (13.15)$$

where $\mathbb{1}\{A\}$ is the indicator of event A :

$$\mathbb{1}_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (13.16)$$

and $\mathbf{x}^i \leq \mathbf{t}$ meaning $x_j^i \leq t_j$, $j = 1, \dots, d$. The last relation defines a partial order and if it is true, then \mathbf{x}^i is either dominated by \mathbf{t} or equal to it.

The estimation of Value-at-Risk and Conditional Value-at-Risk by means of the ECDF is explained in the following subsections.

13.4.1.1 Value-at-Risk (Quantile) Estimation Using ECDF

Value-at-Risk for a scalar random variable X at a given confidence level α can be directly computed from Eq. 13.15. Hence, if X_1, X_2, \dots, X_n are n independent and identically distributed observations of the random variable X , then the estimation of the α -VaR of X is given by

$$\hat{v}^{\alpha;n} = X_{[n\alpha]:n} = \hat{F}_n^{-1}(\alpha) \quad (13.17)$$

where $X_{i:n}$ is the i -th order statistic from the n observations, and

$$\hat{F}_n(t) = \sum_{i=1}^n \mathbb{1}\{X_i \leq t\} \quad (13.18)$$

is the empirical CDF constructed from the sequence \tilde{X} of x_1, x_2, \dots, x_n . Note that the hat symbol ($\hat{\cdot}$) indicates estimated quantities.

13.4.1.2 Cumulative Value-at-Risk (Superquantile) Estimation Using ECDF

Regarding the estimation of the superquantile, according to [15], c^α can also be written as a stochastic program:

$$c^\alpha = \inf_{t \in \mathbb{R}} \left\{ t + \frac{1}{1 - \alpha} E[X - t]^+ \right\} \tag{13.19}$$

with $[a]^+ = \max\{0, a\}$. The set of optimal solutions to the stochastic program is $T = [v^\alpha, u^\alpha]$ with $u^\alpha = \sup t : F(t) \leq \alpha$. In particular, $v^\alpha \in T$, so

$$c^\alpha = v^\alpha + \frac{1}{1 - \alpha} E[X - v^\alpha]^+ \tag{13.20}$$

When X has a positive density in the neighbourhood of v^α , then $v^\alpha = u^\alpha$. Under these conditions, the above formula can be also directly derived from Eq. 13.13. So, in the case of finite number of samples, with X_1, X_2, \dots, X_n independent and identically distributed (i.i.d.) observations of the random variable X , the estimation of c^α is given by:

$$\hat{c}^{\alpha;n} = \hat{v}^{\alpha;n} + \frac{1}{n(1 - \alpha)} \sum_{i=1}^n [X_i - \hat{v}^{\alpha;n}]^+ \tag{13.21}$$

13.4.2 Estimation of Risk Functions Using WECDF

It was above stated that the ECDF is a step function that jumps up a fixed quantity, $1/n$, for each data point belonging to the sorted set of samples. Conversely, the WECDF can be considered as a step function that has a variable size jump, w_i :

$$\hat{F}_{\mu; \mathbf{w}}^n(\mathbf{t}) = \sum_{i=1}^n w_i \mathbb{1}\{\mathbf{x}^i \leq \mathbf{t}\} \tag{13.22}$$

with the related constraint

$$\sum_{i=1}^n w_i = 1 \tag{13.23}$$

The formula for VaR estimation starting from a WECDF is a generalization of Eq. 13.18, and requires two steps. Firstly, the k_α index of the sorted sample set has to be chosen according to the following inequalities:

$$\sum_{k=1}^{k_\alpha} w_k \geq \alpha > \sum_{k=1}^{k_\alpha-1} w_k \quad (13.24)$$

then $\hat{v}^{\alpha;n}$ (α -VaR) is simply given by choosing the k_α^{th} element of ordered set:

$$\hat{v}^{\alpha;n}(x) = x_{(k_\alpha)} \quad (13.25)$$

Similarly, $\hat{c}^{\alpha;n}(x)$ (α -CVaR) is given by

$$\hat{c}^{\alpha;n}(x) = \frac{1}{1-\alpha} \left[\left(\sum_{k=1}^{k_\alpha} w_k - \alpha \right) x_{(k_\alpha)} + \sum_{k=k_\alpha+1}^n w_k x_{(k)} \right] \quad (13.26)$$

The use of WECDF becomes important in cases in which the statistical sample has to be corrected or re-elaborated with some post processing steps. This, for example, is the case of importance sampling, where the data set is sampled according to distributions that may differ substantially from those of the underlying random variables. Indeed, one of the possible approaches to the correct input distributions is the assignment of a different weight to each sample. In statistics, this method is called change of probability measure. In this field, several techniques have been developed [3], and, among these techniques, the one based on WECDF is thoroughly described in [1].

13.4.3 Bootstrap Error Analysis

As it was previously mentioned, the results of the robust optimization problem are influenced by the quality of the risk function estimate. A possible approach to deal with this problem is the use of computational statistics methods, like the bootstrap, developed by Efron in 1979 [6]. As a general term, bootstrapping can be defined as an operation that will allow a system to self-generate from its small subsets. Hence, confining the definition to the statistical field, it is a computational re-sampling technique that provides the confidence intervals of statistics without a prior assumption about the type of the distribution function. In this work, it is used to assess the quality of the risk function estimates used in the optimization process.

Given a statistic $T(x_1, x_2, \dots, x_n)$ evaluated on a set of data $\{x_1, x_2, \dots, x_n\}$, the method consists of the following steps:

- Forming new sample sets $\{x_1^*, x_2^*, \dots, x_n^*\}$, also known as bootstrap samples, of the same size of the real sample by performing a random selection of the original observation with replacement. Usually, the same observation is introduced several times in the bootstrap samples.

- Then, the statistic of interest $T(x_1^*, x_2^*, \dots, x_n^*)$ is calculated for these new samples.

This statistic will show a probability distribution of its own. Thus, from this distribution, the confidence intervals of the risk functions, like VaR or CVaR, are obtained. In other words, the evaluation of confidence intervals would require repeated samples of a given population, but only one sample is available. Thus, the bootstrap method treats the real sample as a population and the repeated samples needed for confidence interval estimation are obtained by re-sampling it with replacement.

Finally, it must be mentioned that, although very attractive for its simplicity, the bootstrap technique has also several disadvantages, thoroughly discussed in scientific literature. Maybe the main drawback is that the bootstrap samples are related to the original (real) sample in the same manner that the original sample is related to the unknown population. Hence, if the original population sample is not sufficiently representative of the whole population features, then the confidence intervals computed by bootstrap might be completely misleading.

13.5 Application Example

This section is aimed to give a simple but significant example of robust aerodynamic design optimization problem focused on an airfoil in incompressible conditions subject to geometric and aerodynamic constraints. The goal is the improvement of the airfoil performance by changing its shape. When a robust version of this problem is faced, an optimal solution that is less vulnerable with respect to uncertainties in operating conditions and geometric shape is obtained. The baseline airfoil is the NACA 2412. The design conditions assumed are Mach = 0 and Reynolds = 0.5×10^6 .

The airfoil performance is measured by a quantity of interest Q defined by drag coefficient c_d plus some constraints that are here considered as penalties. Consequently, the robust optimization problem requires the minimization of the α -CVaR of Q , with α set to 0.9. The equality constraints are the lift coefficient (c_l), which is fixed to 0.5, and the maximum thickness (t), which is fixed to the 12% of the airfoil chord (c). The inequality constraints are the trailing edge angle (TEA), which must be greater than or equal to 13° , the leading edge radius (LER), that must be greater than or equal to 0.7% of the chord, and the boundary layer transition point on the airfoil lower surface (XTR_{LOW}) that cannot be located at x/c greater than 0.95. A constraint on the pitching moment was not considered. In addition, an ERROR variable is set to 1 when the solver does not converge. In summary, the problem constraints are reported below:

$$\begin{cases} c_l = 0.5 \\ t/c = 0.12 \\ XTR_{LOW} \leq 0.95c \\ TEA \geq 13^\circ \\ LER \geq 0.007c \end{cases} \quad (13.27)$$

Hence, the robust optimization problem is formulated as:

$$\min_{z \in Z \subseteq \mathbb{R}^n} CVaR(Q) \quad (13.28)$$

with

$$Q = c_d + p^+(TEA, 13^\circ) + p^+(LER, 0.007c) \quad (13.29)$$

In this case the constraints regarding the leading edge radius and the training edge angle are treated as quadratic penalties:

$$p^+(x, y) = \begin{cases} 0 & \text{if } x \geq y \\ (x - y)^2 & \text{if } x < y \end{cases} \quad (13.30)$$

Instead, the constraints on the c_l and on the thickness do not appear because they are automatically satisfied by the computation procedure by changing the angle of attack and by re-scaling the airfoil thickness to the assigned value.

The robust optimization problem is built by introducing uncertainties in the airfoil section shape that is parametrized as a linear combination of an initial geometry $(x_0(s), y_0(s))$, and some modification functions $y_i(s)$. Moreover, to describe geometry uncertainties, further $z_j(s)$ modification functions are introduced. So, the airfoil shape, including uncertainties, is described by

$$x(s) = x_0(s), \quad y(s) = k \left(y_0(s) + \sum_{i=1}^n w_i y_i \right) + \sum_{j=1}^m U_j z_j \quad (13.31)$$

where the airfoil shape is controlled by the design parameters w_i and by the scale factor k . The uncertainty on shape and thickness of the airfoil is described by the U_j random variables. In this optimization problem 20 uniform random variables, in the range $[-0.1, 0.1]$, have been used. Moreover, the population is generated by means of a Monte Carlo algorithm and it has size equal to 100. It is important to note that the airfoil is rescaled to the assigned thickness before the application of the random variables that describe the uncertainty in shape.

The performances of the parametric airfoil obtained by using Eq. 13.31 are computed by an aerodynamic analysis code, namely Prof. Drela's XFOIL code [5]. It is based on a second order panel method interactively coupled to a boundary layer

Table 13.1 Optimization parameters for the optimization

Maximum evaluations	Population size	Initial standard deviation
7000	20	0.1

integral module. Moreover, the laminar to turbulent flow transition is predicted using the e^N method [17].

The optimization algorithm selected for solving the described design problem is the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [8], which is a stochastic optimization algorithm based on self-adaptation of the covariance matrix of a multi-variate normal distribution. It is mainly used for design optimization problems up to a few hundreds of design variables. The parameters used for the optimization algorithm are the maximum number of allowed evaluations, the population size λ , and the initial standard deviation σ . The parameters set for this problem are reported in Table 13.1.

Furthermore, it was mentioned that in case of non-convergence of the solver an ERROR flag was set equal to 1. When robust optimization is faced, the treatment of these cases is crucial for the optimization process. The quantity of interest to be minimized, $CVaR$, depends on the upper tail of the Cumulative Distribution Function. As a consequence, assigning a high value to the objective in the cases where convergence is not achieved implies a too high $CVaR$ value that could be detrimental for the optimization algorithm behaviour. Hence, a proper value of the objective in these cases must be decided. In particular, in this optimization problem, the worst objective value selected between the properly converged cases, is assigned to those in which the error flag is set. Numerical tests lead to conclude that this was the setup with lowest impact on the optimization process behaviour.

13.5.1 Results

The obtained results are here commented and compared with the baseline airfoil and the equivalent deterministic solution of the optimization problem. Firstly, in Fig. 13.3, the airfoil shape of the robust optimized airfoil (dotted line) is compared with the baseline airfoil (solid line) and the deterministic optimized airfoil (dashed line).

In addition, the Cumulative Distribution Function obtained by introducing uncertainties in the airfoil shape are reported in Fig. 13.4 for the initial NACA 2412 airfoil (solid line), for the deterministic optimized airfoil (dashed line), and for the robust optimized airfoil (dotted line).

The comparison of the CDFs related to the deterministic and robust optimized airfoils highlights that the robust optimal solution is less vulnerable to uncertainties in geometric shape with respect to the deterministic one. This can be also deduced

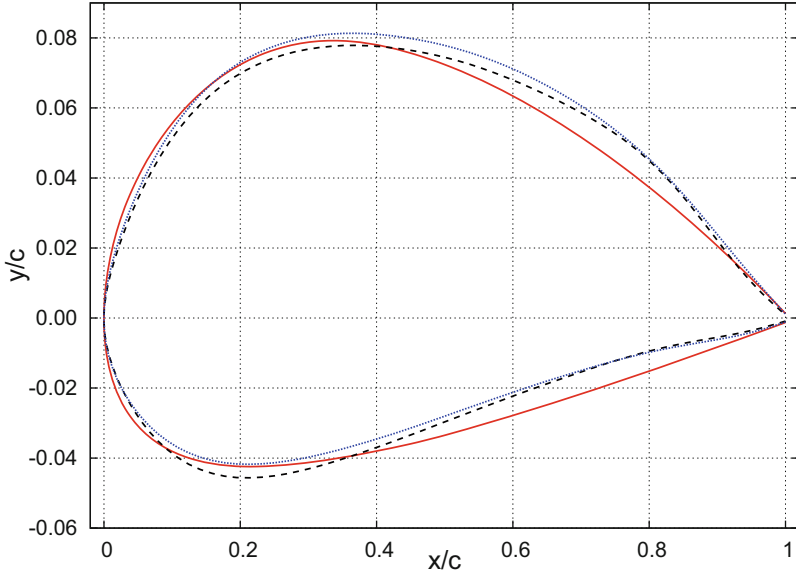


Fig. 13.3 Airfoil shape comparison of the robust optimized airfoil (\cdots), deterministic optimized airfoil ($---$) versus the baseline NACA 2412 airfoil ($---$)

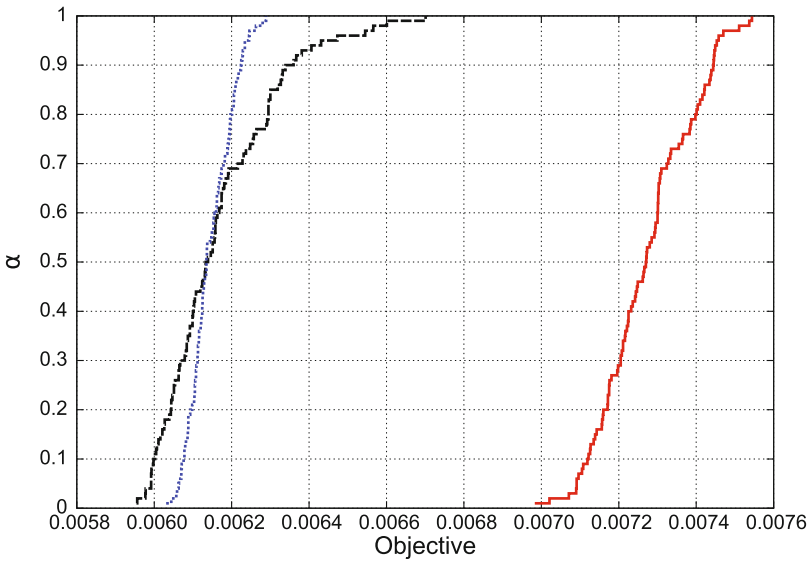


Fig. 13.4 CDF obtained by the variation in airfoil shape using the robust optimized airfoil (\cdots), the deterministic optimized airfoil ($---$), and the baseline NACA 2412 airfoil ($---$)

Table 13.2 Risk measurement based on the obtained cumulative distribution functions

	$0.9 - VaR \cdot 10^4$		$0.9 - CVaR \cdot 10^4$
Baseline NACA 2412 airfoil	74.5		75.0
Deterministic optimized airfoil	63.4		64.8
Robust optimized airfoil	62.2		62.5

by the observation of the value Value-at-Risk and Conditional Value-at-Risk with $\alpha = 0.9$ provided in Table 13.2.

References

1. S. Amaral, D. Allaire, K. Willcox, Optimal L_2 -norm empirical importance weights for the change of probability measure. *Stat. Comput.*, 1–19 (2016)
2. P. Artzner, F. Delbaen, J.-M. Eber, D. Heath, Coherent measures of risk. *Mathematical Finance* **9**(3), 203–228 (1999)
3. P. Billingsley, *Probability and Measure*, 3rd edn. (Wiley, New York, NY, 1995)
4. J.K. Blitzstein, J. Hwang, *Introduction to Probability* (Chapman and Hall/CRC, London, 2014)
5. M. Drela, H. Youngren, *XFOIL 6.94 User Guide* (MIT Aero & Astro, 2001)
6. B. Efron, Bootstrap methods: another look at the jackknife, in *Breakthroughs in statistics* (Springer, New York, 1992), pp. 569–593
7. G. Gori, M. Zocca, G. Cammi, A. Spinelli, P.M. Congedo, A. Guardone, Accuracy assessment of the Non-Ideal Computational Fluid Dynamics model for siloxane MDM from the open-source SU2 suite. *Eur. J. Mech. B Fluids* **79**, 109–120 (2020)
8. N. Hansen, The CMA evolution strategy: A comparing review, in *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, ed. by J.A. Lozano, P. Larrañaga, I. Inza, E. Bengoetxea (Springer, Berlin, Heidelberg, 2006), pp. 75–102
9. D.S. Lee, L.F. Gonzalez, J. Périaux, K. Srinivas, Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. *IEEE Trans. Evol. Comput.* **15**(2), 133–150 (2011)
10. M. Padulo, M.S. Campobasso, M.D. Guenov, Novel uncertainty propagation method for robust aerodynamic design. *AIAA J.* **49**(3), 530–543 (2011)
11. G.-J. Park, T.-H. Lee, K.H. Lee, K.-H. Hwang, Robust design: An overview. *AIAA J.* **44**(1), 181–191 (2006)
12. D. Quagliarella, E. Iuliano, Robust design of a supersonic natural laminar flow wing-body. *IEEE Comput. Intell. Mag.* **12**(4), 14–27 (2017)
13. D. Quagliarella, Value-at-risk and conditional value-at-risk in optimization under uncertainty, in *Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013–2016) Funded by the European Union*, ed. by C. Hirsch, D. Wunsch, J. Szumbariski, Ł. Łaniewski-Wołk, J. Pons-Prats (Springer International Publishing, Cham, 2019), pp. 541–565
14. D. Quagliarella, E.M. Tirado, A. Bornaccioni, Risk measures applied to robust aerodynamic shape design optimization, in *Flexible Engineering Toward Green Aircraft* (Springer, New York, 2020), pp. 153–168
15. A.A. Trindade, S. Uryasev, A. Shapiro, G. Zrazhevsky, Financial prediction with constrained tail risk. *J. Bank. Finance* **31**(11), 3524–3538 (2007)
16. A.W. van der Vaart, *Asymptotic Statistics* (Cambridge University Press, Cambridge, 1998)
17. J. Van Ingen, The eN method for transition prediction. Historical review of work at TU Delft, in *38th Fluid Dynamics Conference and Exhibit*, p. 3830 (2008)

18. G.N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design: with Applications* (McGraw-Hill, 2001)
19. D. Xiu, G.E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos. *J. Comput. Phys.* **187**(1), 137–167 (2003)
20. C.-S. Yu, H.-L. Li, A robust optimization model for stochastic logistic problems. *Int. J. Prod. Econ.* **64**(1–3), 385–397 (2000)
21. T.A. Zang, *Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles* (National Aeronautics and Space Administration, Langley Research Center, 2002)

Chapter 14

Best Practices for Surrogate Based Uncertainty Quantification in Aerodynamics and Application to Robust Shape Optimization



Christian Sabater

Abstract This chapter introduces the use of aerodynamic shape optimization applied to industrial problems, motivates the use of a robust approach over the classical deterministic optimization, and presents different alternatives for the robust-based and reliability-based problems. The use of surrogates for the Uncertainty Quantification of operational and geometrical uncertainties is a cost-effective solution for high dimensional models if the gradient information is introduced by means of the adjoint method. Finally, the proposed methodology is applied through the reliability-based optimization of an airfoil under operational uncertainties.

Keywords Aerodynamic robust design · Quantile optimization · Surrogate based Uncertainty Quantification · CFD

14.1 Introduction

Aerodynamic shape optimization, the improvement the aerodynamic aircraft performance by modifying its external shape, plays a key role to reduce aircraft direct operating costs, noise and emissions of the greenhouse gases [1]. This is an active field of research linked to improvements in Computational Fluid Dynamics and the development of high performance computing capabilities for numerical methods.

C. Sabater (✉)

German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology,
Braunschweig, Germany

e-mail: Christian.SabaterCampomanes@dlr.de

© Springer Nature Switzerland AG 2021

M. Vasile (ed.), *Optimization Under Uncertainty with Applications to Aerospace Engineering*, https://doi.org/10.1007/978-3-030-60166-9_14

14.1.1 *Deterministic Optimization*

A typical optimization setup aims to obtain the “best” solution to a given problem. This “best” solution does not need to be the global optimum of the problem, but in most cases it being sufficiently superior is enough, a local optimum that satisfies the design requirements and constraints. In the case of aerodynamic shape optimization problems, due to the complex nature of computational fluid dynamics, the optimization must be performed within realistic run times according to the computational resources. Usually the objective is to minimize the drag coefficient of an aircraft under several constraints at given operating conditions.

$$J^*(X^*) = \min_X C_D(X, A_0) \quad (14.1)$$

where C_D is the drag coefficient depending on the design variables X at a given operating conditions A_0 . These are usually the Mach number, M_0 , Reynolds number, Re , and lift coefficient, C_L .

However, Eq. (14.1) is not an accurate representation of reality, especially if the geometry or operating conditions are not fixed and are subject to uncertainty.

14.1.2 *Motivation of Robust Design*

Traditionally, the use of simulation-based design optimization in aerodynamics has been carried out in a deterministic fashion, neglecting uncertainty. In this case both the design variables (shape parameters) and operating conditions (Mach number and lift coefficient) are fixed in each iteration of the optimization process.

However, the sensitivity of the final solution to small changes in the wing geometry due to manufacturing tolerances can affect the real performance of the aircraft. In this case, tightening the manufacturing tolerances may not be a feasible solution due to the increase in the production cost. It is also impossible to maintain the same shape during the flight operations due to wear and tear. Also, fluctuations in the operating conditions such as the Mach number (velocity, air density, temperature) or lift coefficient (change in aircraft weight) cannot be avoided.

In practice, deterministic optimization can result in serious performance losses when accounting for the uncertainty. The main problem is the weak problem formulation of Eq. (14.1). The different trade-offs between the design parameters at different operating conditions must be considered through a robust optimization method. As a result, it is necessary to directly assess the effect in the objective function of relatively small aleatory (or irreducible) uncertainties [2].

14.2 Robust Design Approaches for Aerodynamic Shape Optimization

Historically, several approaches have been available for robust aerodynamic shape optimization. In this section, they are briefly introduced.

14.2.1 Multi-Point Optimization

Multi-point optimization is the most widely used approach in aerodynamic shape optimization to reduce the sensitivity to the operating conditions. The optimization takes into account discrete points through the replacement of the original formulation of the objective function (evaluated at one condition), by a weighted average of m cost functions at given operating conditions A_i . This is commonly used to reduce the drag at both landing and cruise or at a given Mach range.

$$J^*(X^*) = \min_X \sum_{i=1}^m w_i C_D(X, A_i) \quad (14.2)$$

where w_i are weights given to the objective function at a given operating conditions A_i . The main disadvantage is the strong point optimization effect [3]. The optimum configuration strongly depends on the chosen operating conditions, and its performance is usually worse at intermediate ones that are completely ignored by the optimizer. A possible solution is to make the number of operating conditions larger. Another alternative, in case that the optimum configuration is expected to operate at a given Mach interval, is to randomly choose the operating conditions [3].

14.2.2 Worst-Case Approach

It consists of the determination of a geometry whose maximum drag is minimum. It is solved by the Min–Max approach, as in each optimization iteration the maximum drag (worst case) must be solved first.

$$J^*(X^*) = \min_X \max_{A_i} C_D(X, A_i) \quad (14.3)$$

where A_i represent the different operating conditions. These can either be deterministic or follow a probabilistic distribution. For a given geometry, the drag coefficient must be evaluated at all the possible operating conditions, in order to obtain its maximum value (that will be minimized). The optimization does not take into account the fluctuations of the objective function following the geometrical or

operational uncertainties. It only focuses on the operating condition with larger drag (usually at high Mach and/or lift coefficients). As a result of not taking into consideration the global behavior but the worst-case scenario, it is more appropriate for Structural Optimization field rather than for Aerodynamic Shape Optimization problems. In this case, a very conservative design will be obtained. For example, the solution can be optimal with respect to very high Mach numbers, while during the rest of the flight envelope the opportunity of further reducing drag is lost, leading towards an increase in fuel consumption. In addition, this approach is dependent on the choice of operating condition A_j .

14.2.3 Interval Analysis

This methodology is used if the uncertainty in the parameters is recognized but cannot be quantified in statistical terms. It is a two objective optimization problem focused on the minimization of the median of the objective function value and the extent of its interval [4].

$$\min_X \begin{cases} \frac{C_{D_{max}}(X) + C_{D_{min}}(X)}{2} \\ C_{D_{max}}(X) - C_{D_{min}}(X) \end{cases} \quad (14.4)$$

This case is similar to the worst-case approach: it requires also a Min–Max approach, and it takes into consideration only extreme events, instead of the variation of the objective function within the interval.

14.2.4 Statistical Approach

In the statistical approach the operating conditions and geometrical uncertainties are modelled as random variables, whose statistical moments (mean, standard deviation, etc.) are defined. Instead of fixed flight conditions A_i , they are random variables ξ . As a result the objective function also follows a random variable.

14.2.4.1 Characterization of Input Uncertainty

The characterization of the random variables follows either engineering empirical experience or gathering of experimental data. When the mean and standard deviation are known, the most common approach is to characterize them as Gaussian Distributions following the maximum entropy theory [3]. In case that only the upper and lower bounds are known, a uniform distribution is preferred [3].

It is also common to characterize input uncertainties as Beta distributions [3], as they offer the flexibility of representing either truncated Gaussian Distributions (symmetric beta distributions), uniform distributions, and non-symmetrical beta distributions. This is attained by increasing or decreasing the relative importance of the uncertain parameter along the desired interval through its probability distribution function.

14.2.4.2 Definition of Objective Function (I), Robust Design

Robust optimization techniques deal with the influence of operative fluctuations on the overall design, assuming no catastrophic failures. In this case the main goal is to obtain an optimum design that produces the best performance for all possible combinations of the uncertain operating conditions. It is equivalent to minimizing the performance loss due to the uncertainties in the geometry or operating conditions.

Following the Von Neumann–Morgenster decision theory [5], the design with the lowest expected value of the objective function should be chosen. This is also known as the Maximum Expected Value criterion:

$$J^*(X^*) = \min_X \mu_{C_D} = \int_{\xi} C_D(X, \xi) PDF_{\xi}(\xi) d\xi \quad (14.5)$$

This approach is equivalent to the multi-point, where the weights w_i are substituted by the probability density function PDF_{ξ} .

Not only the minimum expected value of the drag is usually considered, but also its variability with respect to the variation of the uncertain parameters. A second criterion commonly added is based on the minimization of the variance of the drag coefficient:

$$J^*(X^*) = \min_X \sigma_{C_D}^2 = \int_{\xi} (C_D(X, \xi) - \mu_{C_D})^2 PDF_{\xi}(\xi) d\xi \quad (14.6)$$

A common approach of the trade-off between the expected value and the variability of the objective function is the minimization of the weighted sum of mean and standard deviation:

$$J^*(X^*) \min_X w_0 \mu_{C_D} + w_1 \sigma_{C_D}^2 \quad (14.7)$$

As a result, it is possible to consider not only uncertainties (random fluctuation) of the operating conditions and geometry, but also the frequency of occurrence, following the PDFs. Events with the highest probability of occurrence will have a larger influence in the optimization process. Instead of weights, the non-dominated solutions can be obtained and shown in a Pareto chart.

14.2.4.3 Definition of Objective Function (II) Reliability Based

Another popular formulation follows from the minimization of the 90% or 95% quantile function (also called inverse cumulative distribution function), of the drag coefficient by means of the Empirical Cumulative Distribution Function (ECDF) [6]. This approach has been previously implemented in optimization under uncertainty problems for aerodynamic shape optimization [6, 7]. It is also called reliability-based approach.

$$J^*(X^*, \xi) = \min_{X, \xi} C_{D_{95\%}}(X, \xi) \quad (14.8)$$

It is equivalent to minimizing the maximum drag that can be obtained with a certain probability. In contrast to Min–Max problems that are too conservative, this approach allows for a certain margin by accounting for the probability of occurrence.

The quantile is obtained from the inverse of the ECDF.

$$\text{ECDF}_n(C_D) = \frac{\text{number of elements in sampling } \leq C_D}{n} = \frac{1}{n} \sum_{i=1}^n I(C_{D_i} \leq x) \quad (14.9)$$

where n is the number of sampling points and I is the indicator function defined as:

$$I(C_{D_i} \leq C_D) = \begin{cases} 1, & \text{if } C_{D_i} \leq x \\ 0, & \text{if } C_{D_i} > C_D \end{cases} \quad (14.10)$$

Following the Glivenko–Cantelli theorem [8], the ECDF converges to the CDF when the number of samples is large enough.

This definition of the objective function (Eq. (14.8)) is similar to that of the worst-case approach presented in Eq. (14.3). The difference is that the robust objective function includes a probabilistic formulation and the worst-case accounts for the maximum drag. The optimum configuration is strongly conditioned by the selection of the quantile [9]. It is possible to perform a multi-objective optimization with different quantiles as objective functions [6].

14.2.4.4 Evaluation of Statistics

A limitation on the implementation of Eqs. (14.5), (14.6), and 14.9 is that in each optimization iteration, a full numerical integration of the statistics of the Quantity of Interest is required through Monte Carlo or Quasi Monte Carlo. This is a problem because the determination of the drag through Computational Fluid Dynamics (CFD) is computationally expensive. As a result, several alternatives are available, such as the analytical approximation through Taylor series expansion

of the statistical moments, the use of sparse sampling techniques or the use of approximations of the cost function.

The use of surrogate models, approximations of the CFD solution that can be cheaply evaluated, has acquired importance in the last years for aerodynamic shape optimization [4, 7], and will be explained with more detail in the following section.

14.3 Surrogate Models for Uncertainty Quantification

This section introduces the use of surrogate models for robust aerodynamic shape optimization with special focus on Uncertainty Quantification. The benefits of surrogate models with respect to other uncertainty quantification methods are outlined. Finally, Gradient-Enhanced Kriging surrogate technique is shown to break the curse of dimensionality in situations where the gradient information is cheaply available.

14.3.1 Surrogate Models Overview

In modern engineering design problems, the use of expensive simulations makes unfeasible the full exploration of the design space. It is not always possible to analyze all the competing options, especially in the aerospace industry, where the modelling of fluid dynamics through numerical methods is one of the most computationally demanding fields [4, 10]. When dealing with CFD based optimization, a deterministic solution is already a computationally expensive approach. In case of robust aerodynamic shape optimization, i.e. with optimization under uncertainty, the high cost of quantifying the uncertainty in each iteration of the optimization process makes necessary the use efficient uncertainty quantification methods, especially for high dimensional problems.

Surrogate models, also called response surfaces or meta-models, are approximations of expensive high fidelity models that represent the physical quantity of interest (for example, drag, weight or cost), as a function of the design or uncertainty parameters. In the case of aerodynamic shape optimization, usually surrogate models of the drag are built as a function of the geometrical parameters. These continuous, multi-dimensional models based on sampled data, require of a one-time upfront investment in order to get instant, online evaluation of the data.

Surrogate models have effectively been applied in global aerodynamic shape optimization problems [11] following an iterative process by balancing exploitation and exploration of the design landscape. Due to the capability of approximating the quantity of interest, another possibility is to use surrogate models as an attractive non-intrusive method to perform uncertainty quantification, especially in the context of Computational Fluid Dynamics.

14.3.1.1 Design of Experiments

One of the key elements for the construction of surrogate models is the initial sampling strategy, the Design of Experiments (DoE). Both for optimization and uncertainty quantification, the choice of sampling strategy must follow from:

- The design landscape is fully explored.
- The sampling must follow minimum discrepancy, i.e. the sampling plan must be based on uniform partitions of the unit interval but not regular.
- The projections of the sampling points to the axis of each variable are uniform.
- The number of expensive simulations/full order model evaluations to achieve this complete exploration should be minimum.
- The number of samples should not be fixed. The addition of more points can be used to further refine the surrogate without the need of recomputing the previous ones.

One of the most common sampling approaches is the use of Latin Hypercube Sampling developed by McKay [12]. However, Quasi Monte Carlo techniques that employ deterministic low discrepancy sequences [13] can prove to be more effective to obtain an even sampling distribution [14].

A Sobol Sequence is a low discrepancy, quasi-random sequence that uses a base of two to successively create fined uniform partitions [15]. As a result, it is possible to improve its accuracy as more samples are added, and it is possible to reuse the existing points.

The surrogate accuracy can be improved for uncertainty quantification if the DoE follows the distribution of the input uncertainties [7, 16, 17]. Sampling points should then be added in areas that are prone to be sampled along the mean, rather than in the tails of the input uncertainties. When direct integration is performed on the surrogate, the accuracy will increase in regions that were sampled more often.

14.3.1.2 Refinement Strategy for Uncertainty Quantification

Commonly, in surrogate based optimization, the refinement strategy follows an equilibrium between exploration and exploitation. In the first one, the location of the maximum expected improvement [18] or maximum surrogate error is sampled. In the latter stage, regions around the current optimum are investigated by validating the surrogate optimum in the full order model or through trust-region methods. For a more details on a refinement strategy for optimization, refer to [19].

In a similar manner, it is possible to refine surrogate models to be applied in uncertainty quantification. In case the objective function follows Eq.(14.8), the reliability-based approach, refinement along the 95% quantile can be achieved as in standard optimization, by means of expected improvement [2, 20].

Otherwise, the refinement strategy follows from a global exploration of the uncertain landscape. The most common criteria is sampling the location with the maximum product of the surrogate error with the probability density function of the

Table 14.1 Suitability of surrogate models for Uncertainty Quantification

Surrogate model	Scattered data	Error available	Use of gradient	Convenient for high dimensions
Kriging	Y	Y	Y	Y
Radial basis functions	Y	N	N	Y
Spline	Y	N	Y	N
Polynomial regression	Y	N	Y	N
Sparse grid	N	N	N	N

input uncertainties [21]. This approach focuses on regions that are often sampled through Monte Carlo (addition of input PDFs) and have a high surrogate error. This refinement strategy is in line with the DoE sampling strategy of adding more points in regions that are sampled more often, as eventually the surrogate error is proportional to the distance among the points.

14.3.2 Advantages of Surrogate Modelling for Uncertainty Quantification

Following [10], the engineering requirements for an effective uncertainty quantification technique are:

- Require no more than 100 evaluations of the full order model,
- Scattered evaluation positions can be obtained,
- Possibility to incorporate pre-existing evaluations of the quantity of interest in the uncertainty quantification,
- Robustness towards failure in the evaluation of the objective function (the sample points should not be fixed). This is cannot be possible with fixed integration rules where the sampling distribution is fixed beforehand,
- Availability of the exploitation of error estimation and error reduction techniques,
- Possibility to extend to high dimensional problems.

Based on these points, it is possible to construct Table 14.1 with the suitability of the different surrogate models for Uncertainty Quantification with a special focus on aerodynamic shape optimization problems:

According to these requirements, surrogate methods are an attractive approach for the efficient quantification using CFD. Among the different surrogates, Kriging has become one of the most used ones in engineering [19]. It follows a probabilistic framework according to Bayesian statistics, featuring a built-in measure of the prediction error [19, 22]. In contrast with Polynomial Chaos Expansion, Kriging interpolation works well with nonlinear functions such as the drag coefficient. Finally, by using surrogate models, arbitrary locations of the samples can be chosen in contrast of sparse grid techniques.

14.3.3 Gradient-Enhanced Surrogates for Efficient UQ

When dealing with optimization problems, one of the benefits of surrogate modelling is that the gradients of the objective function are not required. Having available the gradients, it is possible to directly choose a gradient descent approach [23]. However, if a global search is desired, the addition of the gradients can enhance the accuracy of the surrogate model. In the case of surrogate based uncertainty quantification this is crucial, as the addition of gradients can improve the global representation of the design or uncertain landscape for a more accurate integration through classical Monte Carlo Methods.

The main drawback of gradient-enhanced surrogate models is the increase in complexity of the model and the increase in size of the correlation matrix, leading to lengthier surrogate parameter estimation. This is the cost to pay in order to have more accurate predictions.

14.3.3.1 Gradient-Enhanced Kriging

The use of Gradient-Enhanced Kriging (GEK) [19] is especially useful when a large number of uncertain variables is present. Following [10, 20], it is shown that employing GEK using gradient information that is cheaply evaluated, is as effective as common integration methods such as probabilistic collocation and sparse grid techniques. With the increase in dimensionality, it is expected that GEK will be more efficient, as the cost of the gradients is kept constant. The increase in accuracy obtained with GEK w.r.t. Kriging is shown in Fig. 14.1 for an analytical function with 15 samples following Sobol sequences. By obtaining the gradients in addition to the function value, the representation of the landscape is more accurate.

The gradients must only be added to the surrogate if these can be cheaply evaluated. Otherwise, it is more useful to enhance the meta-model by simply evaluating more points of the full order model. The use of traditional methods to obtain the gradients (finite differences or complex step differentiation) should be then substituted by others more advanced such as the adjoint method or algorithmic differentiation.

14.3.3.2 The Adjoint Method: Breaking the Curse of Dimensionality

The adjoint method computes the partial gradients with respect to all design parameters or uncertainties at the cost of about one flow solution; in other words, the cost of the adjoint method is independent of the number of parameters. It provides an effective method to build the Gradient-Enhanced Kriging model of the response surface.

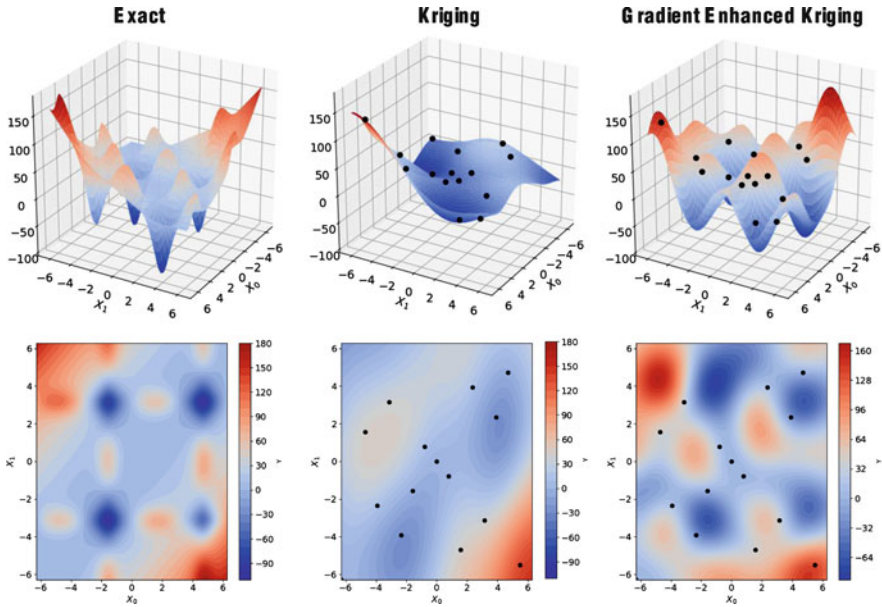


Fig. 14.1 Comparison of Kriging and Gradient-Enhanced Kriging for a given set of sampling points

14.3.4 Computing Statistics on Surrogate Models

Despite the reduced computational cost to evaluate the statistic of interest using surrogate models, the use of Monte Carlo for the evaluation of statistics may be impractical due to its low convergence rate.

A possible alternative is the use of Quasi Monte Carlo techniques [13], as they have a faster convergence rate. The sampling is predefined (derandomized) through Sobol Sequences [14] and the stochastic space is evaluated more efficiently.

14.4 Optimization of RAE2822 Airfoil Under Uncertainty

This section focuses on the application of the concepts previously introduced to the aerodynamic shape optimization under uncertainty of the RAE2822 airfoil at transonic conditions.

14.4.1 Problem Definition

The deterministic and uncertain optimization problems are introduced below. The formulations are based on previous studies focused on the robust design of an airfoil at cruise conditions [2, 7].

14.4.1.1 Deterministic Optimization

The deterministic optimization follows the classical aerodynamic shape optimization formulation and will be compared with the robust optimum. The transonic airfoil RAE2822 whose shape is dependent on the design vector X is optimized for minimum drag C_D at constant operating conditions A_0 . For structural considerations, the airfoil thickness distribution is kept constant by using an implicit constraint in the parametrization.

$$J^*(X^*) = \min_X C_D(X, A_0) \quad (14.11)$$

$$A_0 = \begin{cases} M_0 = 0.76 \\ C_L = 0.5 \end{cases} \quad (14.12)$$

14.4.1.2 Optimization Under Uncertainty

The transonic airfoil RAE2822 is optimized under aleatory operating uncertainties ξ (lift coefficient and Mach number) following a Reliability-Based Design Optimization (RBDO). The 95% quantile of the drag, $C_{D_{95\%}}$ is minimized. From all the possible values of drag due to the uncertainties, this will be smaller or equal to the quantile with a 95% probability.

$$J^*(X^*) = \min_X C_{D_{95\%}}(X, \xi) \quad (14.13)$$

14.4.2 CFD Solver and Numerical Grid

The aerodynamic performance of the airfoil is calculated by solving the Reynolds Average Navier Stokes (RANS) equations with the DLR TAU solver [24]. The selected turbulence model is negative Spalart–Allmaras. The solution is obtained through a 3v multigrid circle, with the lower/upper symmetric Gauss–Seidel implicit method for time integration in backward Euler solver and a central flux discretization. Convergence of the solution is set when the density residual is

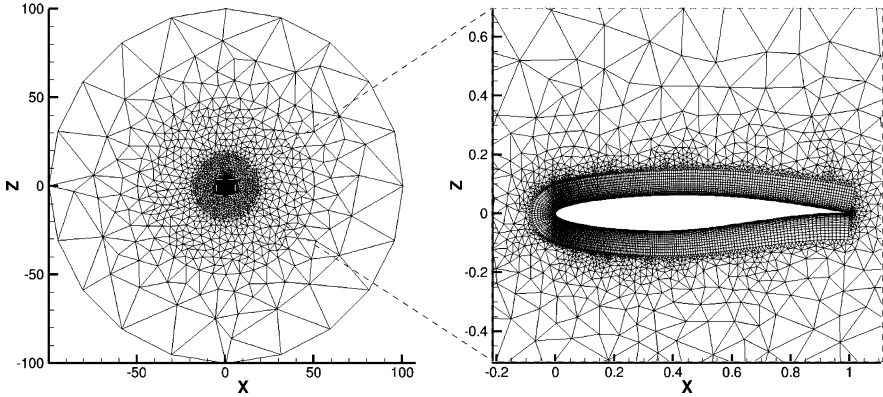


Fig. 14.2 CFD Mesh used in the optimization problem

lower than $1e-8$. The unstructured mesh has 29,000 grid nodes, and is quasi two-dimensional, hybrid with tetrahedral, and prism elements, as shown in Fig. 14.2.

14.4.3 CFD Process Chain

The evaluation of the drag (objective function in deterministic optimization, and used for UQ in the optimization for uncertainty) follows the process of Fig. 14.3.

At each iteration j the geometry is given by the design vector X^j . The change in geometry due to a perturbed design vector is translated to the CFD mesh by means of a radial basis functions mesh deformation tool developed by DLR [25]. The boundary conditions are introduced in the preprocessing and then the solver obtains the solution field. In the postprocessing the aerodynamic forces acting on the surface of the airfoil are integrated in the direction of the incoming flow to obtain the drag coefficient. The complete process chain is handled by Flow Simulator through Python [26].

14.4.4 Parametrization of Deterministic Design Variables

The airfoil camber line is modified by means of five Hicks-Henne bump functions [27]. As a result it is possible to modify the airfoil shape while satisfying the thickness distribution constraint.

The vertical displacement of the chamber line z_i due to the bump function i is controlled by the design variable X_i . A total of five (n_X) design variables X_i compose the design vector X .

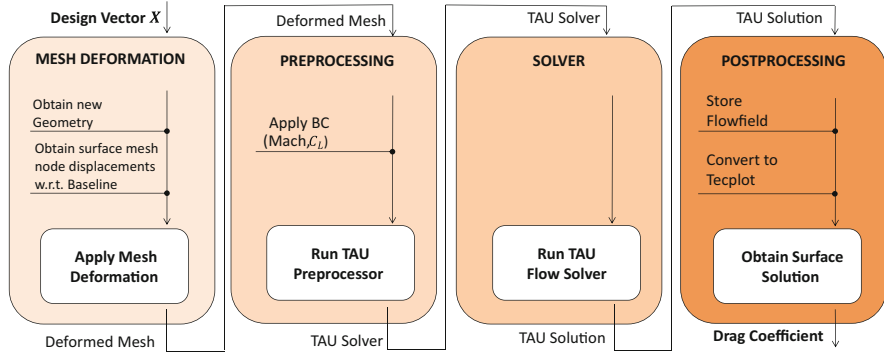


Fig. 14.3 CFD Process Chain used to evaluate the drag coefficient for a given design vector and operating conditions

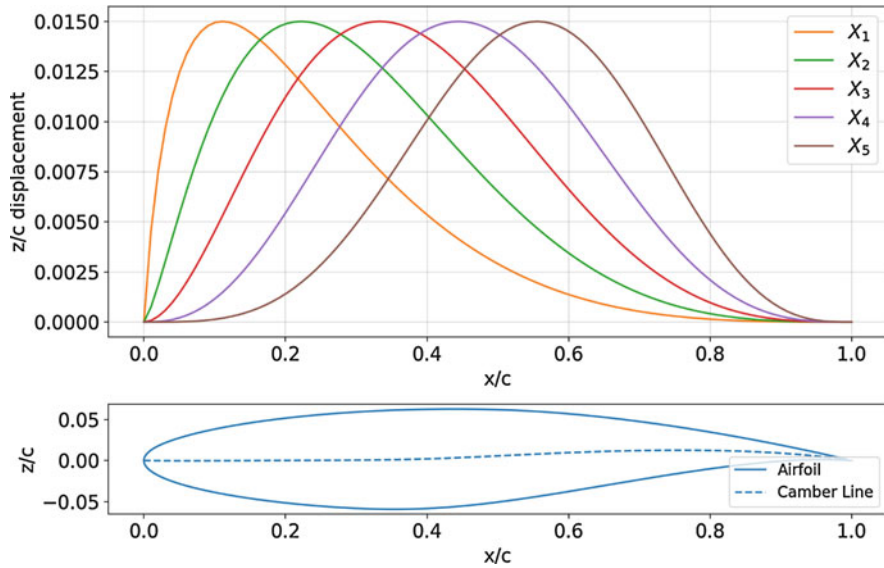


Fig. 14.4 Top: Five Hicks-Henne Bump Function used for the parametrization, in its maximum value $X_i = 0.015$. Bottom: RAE2822 shape and camber line

$$z_i = X_i [\sin (\pi x^m)]^3 \tag{14.14}$$

$$m = \frac{\log(0.5)}{\frac{i+1}{n_X+4}} \tag{14.15}$$

The five Hicks-Henne bump functions along the airfoil are shown in Fig. 14.4. As a result, each of them is responsible for modifying a given section of the airfoil. The five design variables X_i range between $X_i = \pm 0.015$ and are normalized between

Table 14.2 Characterization of operational uncertainties

	Mach number	Lift coefficient
Mean	0.76	0.5
Standard deviation	0.0033	0.00333

0 and 1, corresponding to its minimum and maximum value, respectively. They control the amplitude (positive or negative) of the deformation.

14.4.5 Parametrization of Uncertainties

The operational uncertainties (the freestream Mach number and lift coefficient) present in the day to day of the airfoil operations are considered. These are modelled as symmetric beta distributions with mean and standard deviation following Table 14.2. Values are representative of the uncertainty in the operating conditions as in [7]. The main advantage of a beta distribution is its capability to represent a truncated normal distribution. The truncation facilitates the surrogate based uncertainty quantification by limiting the bounds of the uncertain landscape and surrogate model. The uncertainties are centered around the nominal conditions of the deterministic optimization.

14.4.6 Optimizer

The gradient-free Subplex method of Rowan [28] is used to optimize the objective function. It effectively decomposes the design space into low dimensional subspaces and searches for the convex hull. It is more efficient than the Simplex method [29] by linearly scaling with the dimensionality and can be applied to noisy functions.

14.4.7 Robust Design Framework

At a given optimization iteration j , the characterization of the stochastic space is necessary. As explained in Sect. 14.3.1, to effectively evaluate the statistics of the drag coefficient, a Surrogate Based Uncertainty Quantification approach is followed as shown in Fig. 14.5.

The Surrogate Modelling for Aero-Data Toolbox (SMARTy) developed by DLR is used for the DoE and the construction of the Kriging model [30].

The DoE with 17 samples follows a Quasi Monte Carlo Sobol sequence normalized to the input uncertainties. The number of samples for an accurate uncertainty quantification is determined a priori and validated a posteriori with the optimum configuration with respect to converged Quasi Monte Carlo statistics,

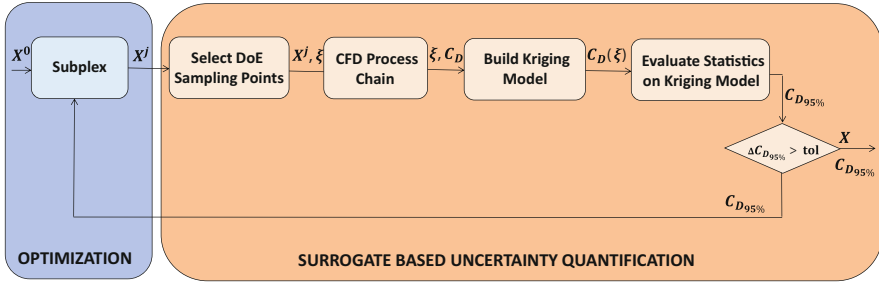


Fig. 14.5 Framework for the Optimization under Uncertainty

as will be shown in Sect. 14.4.8. At each sampling point of the DoE, the drag coefficient with respect to the operational uncertainties ξ at a given design vector X^j is obtained. Then, direct Quasi Monte Carlo integration [13] with 1 million samples is performed in the surrogate to obtain the statistic of interest. From the sampling it is possible to obtain the Empirical Cumulative Distribution Function from Eq. (14.9) and the 95% quantile within the required accuracy.

14.4.8 Validation of the Framework

The Surrogate Based Uncertainty Quantification requires an initial number of sampling points to construct an accurate enough surrogate to evaluate the statistics. This number is obtained before the robust optimization based on the baseline configuration. As shown in Fig. 14.6, the 95% drag quantile is converged with 800 samples with direct integration (Quasi Monte Carlo). This reference is compared to the convergence of the Surrogate Based Uncertainty Quantification. It is shown that 17 samples are enough to construct a surrogate that is able to calculate the reference quantile with an error smaller than 0.5 drag counts. Therefore, 17 samples are used in each iteration of the optimization for the uncertainty quantification stage.

The relevance of the surrogate model is further understood in Fig. 14.7. By using 17 samples for direct integration, the Empirical Cumulative Distribution Function and the 95% quantile is considerably different from the benchmark, obtained with 800 samples. However, if a surrogate model is constructed with those 17 samples (that have been selected following Quasi Monte Carlo) and a full integration is done over the surrogate, the accuracy of the ECDF and quantile reaches the level of the benchmark. It can be seen that the surrogate model improves the accuracy of the Cumulative Distribution Function.

From these results, 17 samples are evaluated using the full order model to perform the surrogate based uncertainty quantification at each iteration of the optimization, under the assumption that the accuracy of the surrogate is kept constant along the optimization. This is validated a posteriori with the robust

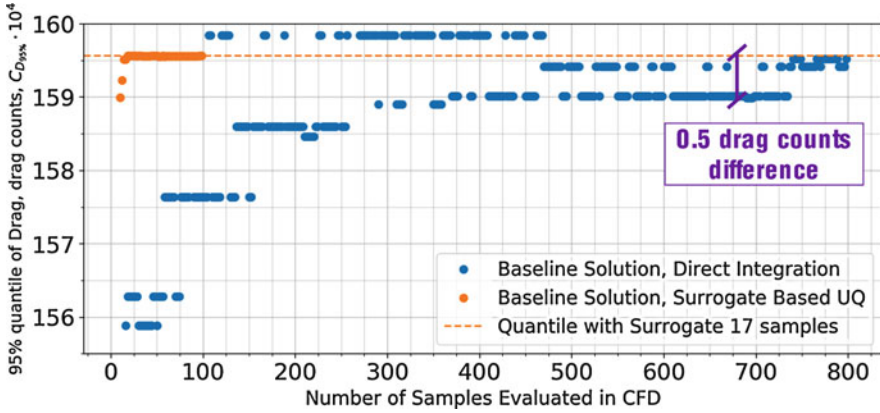


Fig. 14.6 Convergence history of the drag quantile under uncertainties with direct integration, and comparison with surrogate based approach, baseline configuration

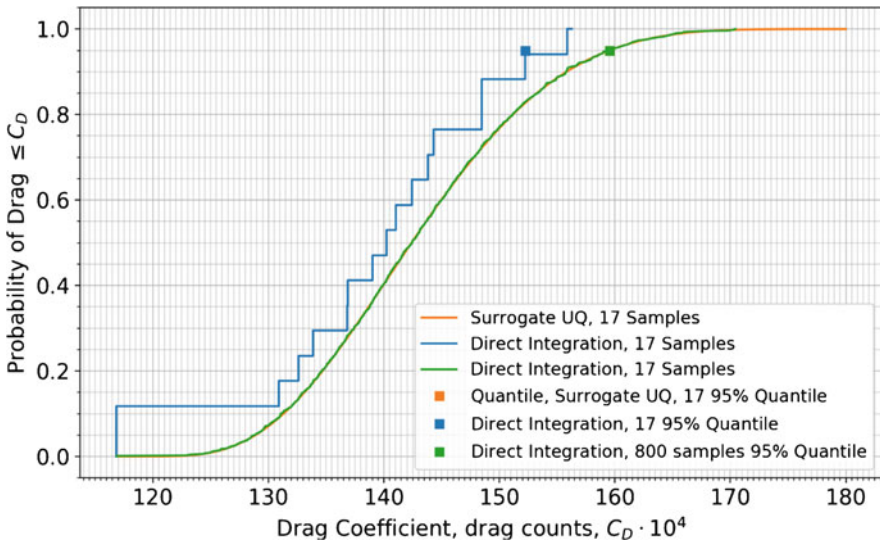


Fig. 14.7 CDF for drag coefficient of the baseline configuration under uncertainties computed with direct integration (17 and 800 samples) and with Surrogate Based approach

optimum configuration, as shown in Fig. 14.8. With 17 samples an error smaller than 0.1 drag counts is obtained when compared with the converged statistics. As a result, the surrogate is able to provide an accurate determination of the statistics along the optimization at a reduced number of function evaluations, validating the proposed approach.

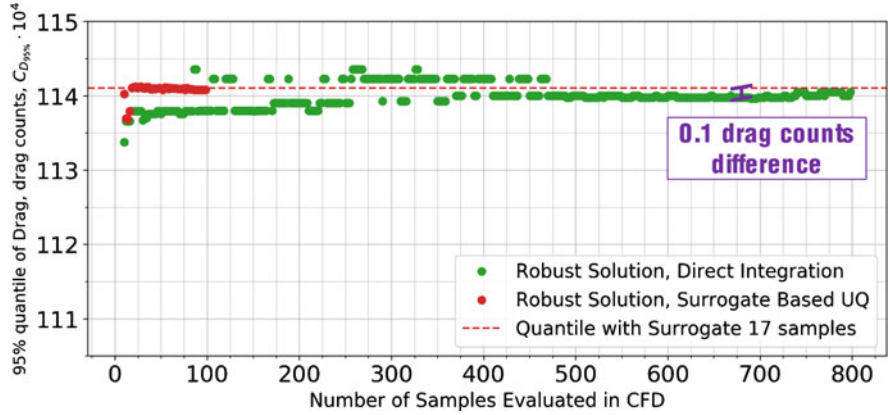


Fig. 14.8 Convergence history of the drag quantile under uncertainties and comparison with surrogate based approach, robust optimum configuration

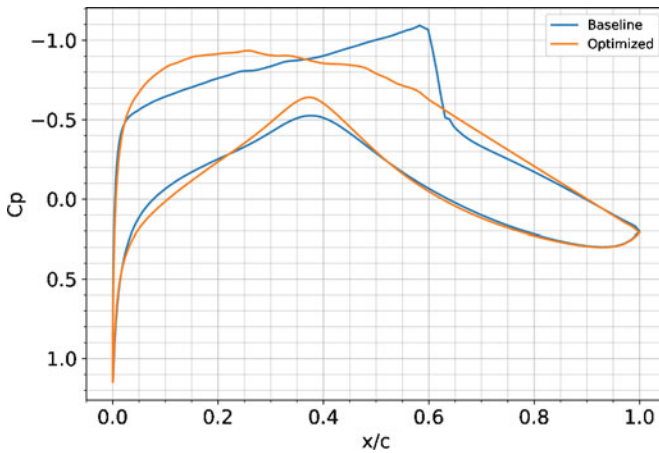


Fig. 14.9 Pressure coefficient for baseline and deterministic optimum configurations

14.4.9 Deterministic Results

The deterministic optimum solution (at nominal operating conditions) is found after 106 evaluations of the objective function (CFD computations). The optimum configuration reduces drag by 22% (31.36 drag counts), from 142.4 to 111.04 drag counts. As shown in Fig. 14.9 from the surface pressure coefficient, the new airfoil shape replaces the normal shockwave of the upper surface by an isentropic compression and reduces the wave drag.

14.4.10 Robust Results

Figure 14.10 shows the convergence history of the optimization under uncertainty. The optimum configuration is obtained after 87 iterations which corresponds to 1479 CFD evaluations, as for each iteration 17 samples are evaluated with CFD to build a surrogate model which in turn is used to evaluate the 95% quantile of the drag coefficient.

The robust optimum configuration reduces the value of the 95% quantile of the drag coefficient by 28.48% (45.77 drag counts).

The airfoils obtained with the robust and deterministic configurations are similar as shown in Fig. 14.13. However, looking at its Cumulative Distribution Function (CDF) and histogram in Fig. 14.11 it is shown that the robust configuration outperforms the deterministic in the 95% quantile as it has a smaller upper tail.

The mean pressure coefficient field represented in Fig. 14.12 with direct integration of 800 samples shows that on average, a strong shock wave is present in the baseline configuration. In the robust optimum configuration the shock wave is dissipated (the pressure increase is more gradual, isentropic) and therefore, the drag is reduced during most of its operating time.

With regard to the standard deviation field, the shock wave location of the baseline configuration is fixed in a small region of the airfoil. However, in that location the standard deviation is higher than of the robust optimum. When taking into account operational uncertainties, the robust optimum presents a shock wave more evenly distributed along the airfoil but with similar strength in all the cases (lower value of standard deviation).

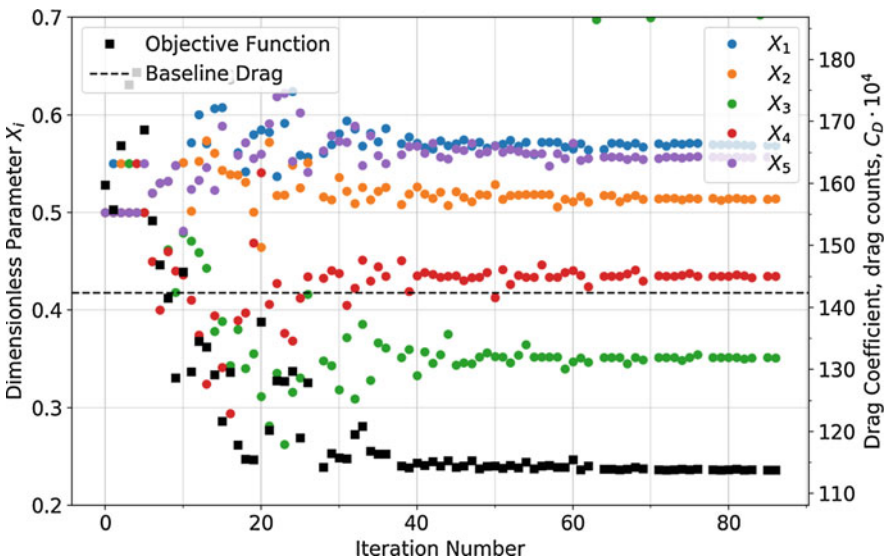


Fig. 14.10 Convergence history of the design parameters and objective function for the Optimization under Uncertainty

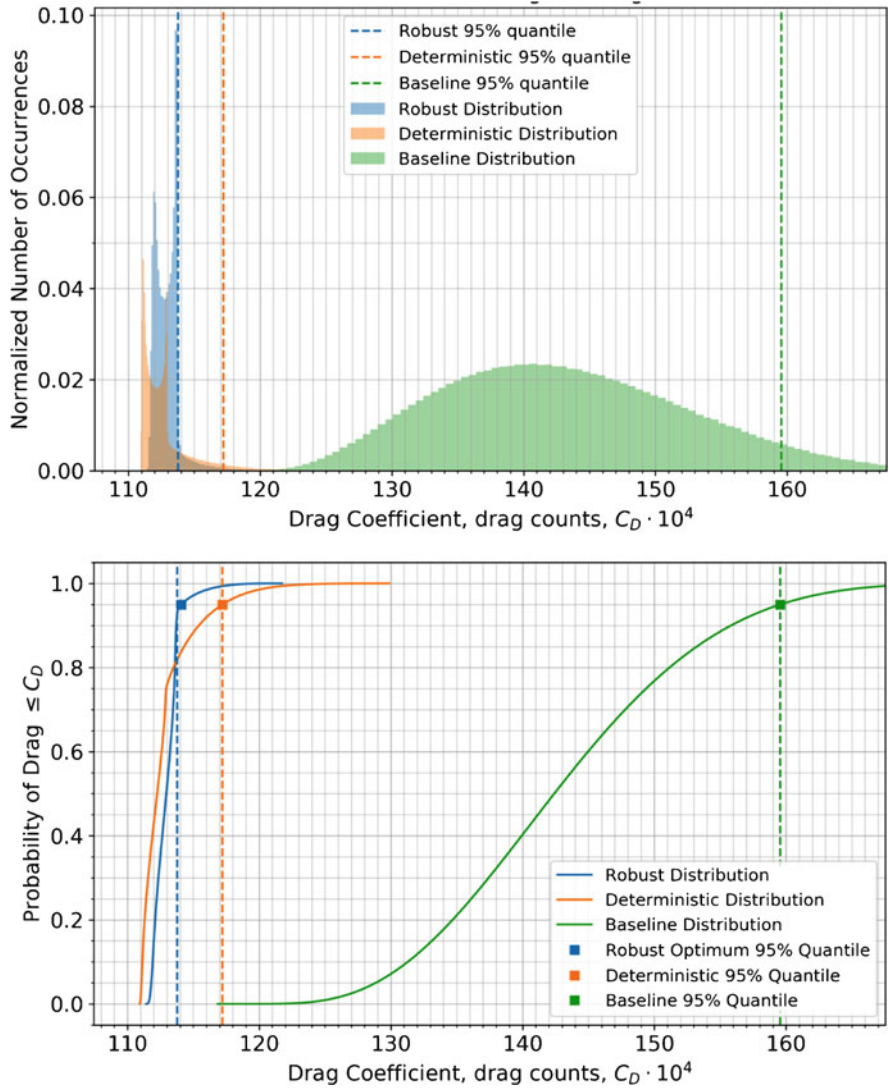


Fig. 14.11 Uncertainty quantification results for baseline, deterministic optimum, and robust optimum. Top: Histogram. Bottom: CDF

It can be concluded that a robust configuration (in terms of drag) is effective in reducing the strength of the normal shockwave while having a larger longitudinal displacement along the airfoil. Opposite to intuition, the robust configuration does not fix the shock wave location but spreads it, in order to reduce drag.

The configurations resulting from the deterministic and robust optimization are shown in Fig. 14.13 together with the baseline RAE2822. In order to avoid the

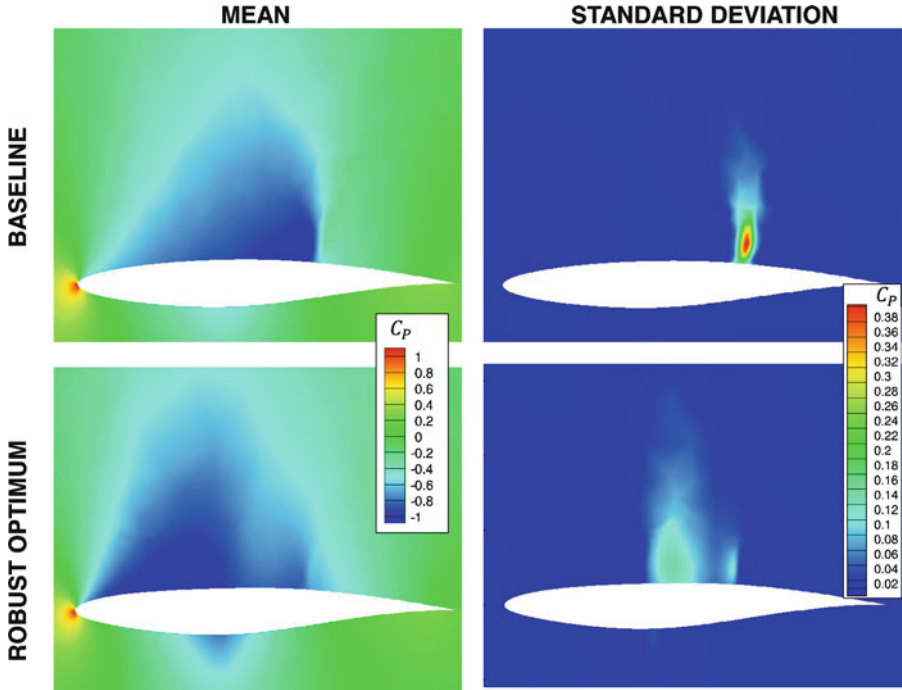


Fig. 14.12 Mean and standard deviation for the pressure coefficient field for baseline and robust optimum configuration

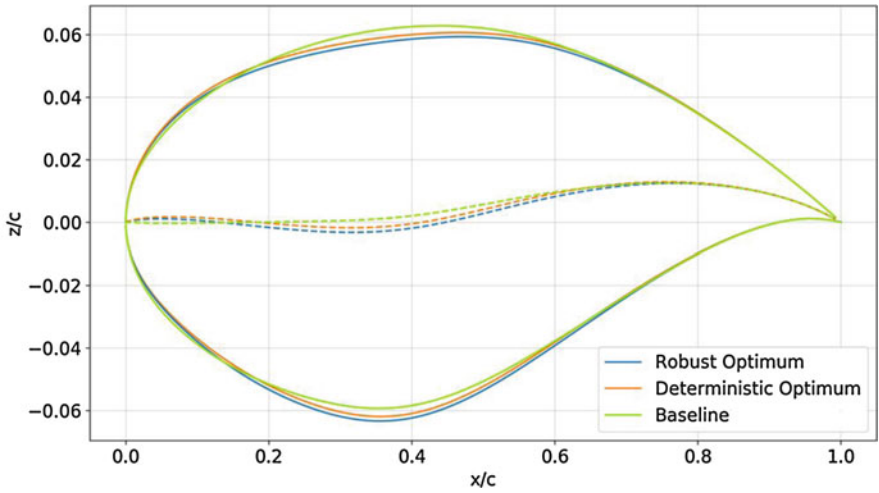


Fig. 14.13 Airfoil shape for baseline, deterministic optimum, and robust optimum configurations

Table 14.3 Comparison of baseline, deterministic optimum, and robust optimum configurations

Configuration	Optimum design vector					Results			
	X_1^*	X_2^*	X_3^*	X_4^*	X_5^*	C_D (nominal)		$C_{D_{98\%}}$	
	Drag counts	Reduction	Drag counts	Reduction	Drag counts	Reduction			
Baseline (RAE2822)	0	0	0	0	0	142.4	-	159.55	-
Deterministic optimum	0.591	0.498	0.418	0.395	0.592	111.04	22.02%	117.22	26.52%
Reliability-based optimum	0.569	0.514	0.352	0.435	0.557	112.5	21.00%	113.78	28.48%

acceleration of the flow and the shockwave at the single design condition, the deterministic configuration decreases the curvature of the upper surface of the airfoil. The robust configuration decreases it even further, in order to mitigate the stronger shockwaves happening at higher lift coefficients and Mach numbers due to the uncertainty. As a trade-off, the drag reduction will not be as good at lower Mach and lift coefficients. As a result, the robust configuration is effective in reducing the upper tail of the CDF of Fig. 14.11.

Table 14.3 summarizes the results of the deterministic and robust optimization. The use of a robust approach further reduces the 95% quantile by 3.5 drag counts, compared to the classical deterministic optimization.

14.5 Conclusions

Traditionally, the use of simulation-based design optimization in aerodynamics has been carried out neglecting uncertainty, in a deterministic fashion. In practice, deterministic optimization can lead to a mismatch between computational and experimental results and serious performance losses due to aleatory uncertainties. The uncertainties in the design conditions must be considered through a robust optimization framework in order to come up with a realistic configuration that is less sensitive to such uncertainties.

To deal with optimization under uncertainty, a statistical approach must be followed in which the operating conditions and geometrical uncertainties are modelled as samples of independent random variables. The objective function becomes either a combination of statistical moments such as mean or standard deviation of a quantity of interest or a quantile. In both cases, a full representation of the stochastic space is necessary to evaluate the uncertainty.

A possibility to reduce the computational cost for an efficient uncertainty quantification is the approximation of the output response through surrogate models. Kriging is especially appropriate for this task as it allows the use of scattered sampling, provides a measure of its prediction error, is convenient for high dimensional problems, and allows the incorporation of gradients through Gradient-Enhanced Kriging. This is specially attractive when the sensitivities can be efficiently evaluated through a discrete adjoint method.

To validate the surrogate based uncertainty quantification for aerodynamic shape optimization, the RAE2822 airfoil is optimized under operational uncertainties. The Subplex optimizer is used and the statistics are evaluated at each iteration using Kriging meta-models. The optimum configuration is able to reduce the 95% quantile of the drag by 28% with respect to the baseline configuration. The optimum configuration is effective in reducing the average strength of the normal shockwave while having a larger longitudinal displacement along the airfoil. The framework is able to provide an accurate estimation of the statistics at a reduced computational cost.

Future work includes the use of more efficient Surrogate Based Optimization methods for the optimization stage, and the combination of the uncertainty and design spaces with the use of gradients in order to further reduce the computational time.

Acknowledgments This work is funded by the European Commission's H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

The author also would like to thank Dr. Daigo Maruyama and Dr. Stefan Goertz for their insight into optimization under uncertainty and surrogate modelling applied to aerodynamic design.

References

1. S.N. Skinner, H. Zare-Behtash, State-of-the-art in aerodynamic shape optimisation methods. *Appl. Soft Comput.* **62**, 933–962 (2018)
2. D. Maruyama, S. Goertz, D. Liu, Robust design measures for airfoil shape optimization, in *Uncertainty Management for Robust Industrial Design in Aeronautics* (Springer, Berlin, 2018), pp. 513–527
3. L. Huysse, Free-form airfoil shape optimization under uncertainty using maximum expected value and second-order second-moment strategies. Techreport 2001-211020, NASA, 2001s
4. R. Duvigneau, Aerodynamic shape optimization with uncertain operating conditions using metamodels. Resreport RR-6143, INRIA, 2007
5. J. Von Neumann, O. Morgenstern, H.W. Kuhn, J. Von Neumann, A. Rubinstein, *Theory of Games and Economic Behavior: 60th Anniversary Commemorative Edition* (Princeton University Press, Princeton, 2009)
6. D. Quagliarella, G. Petrone, G. Iaccarino, Optimization under uncertainty using the generalized inverse distribution function, In *Computational Methods in Applied Sciences* (Springer Netherlands, 2014), pp. 171–190
7. D. Maruyama, D. Liu, S. Goertz, An efficient aerodynamic shape optimization framework for robust design of airfoils using surrogate models, in *Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)* (Institute of Structural Analysis and Antiseismic Research School of Civil Engineering National Technical University of Athens (NTUA) Greece, 2016)
8. R.M. Dudley, *Uniform Central Limit Theorems* (Cambridge University Press, Cambridge, 1999)
9. A.T. Beck, W.J.S. Gomes, Rafael .H. Lopez, L.F.F. Miguel, A comparison between robust and risk-based optimization under uncertainty. *Struct. Multidiscip. Optim.* **52**(3), 479–492 (2015)
10. R. Dwight, Z.-H. Han, Efficient uncertainty quantification using gradient-enhanced kriging, In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (American Institute of Aeronautics and Astronautics, Reston, 2009)
11. A.I.J. Forrester, A. Sabater, A.J. Keane, *Engineering Design via Surrogate Modelling* (Wiley, London, 2008)
12. M.D. McKay, R.J. Beckman, W.J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
13. R.E. Caflisch, Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.* **7**, 1 (1998)
14. S. Kucherenko, D. Albrecht, A. Saltelli, Exploring multi-dimensional spaces: a comparison of Latin hypercube and quasi Monte Carlo sampling techniques (2015). ArXiv e-prints
15. I.M Sobol, On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.* **7**(4), 86–112 (1967)

16. D. Maruyama, S. Goertz, D. Liu, General introduction to surrogate model-based approaches to UQ, in *Uncertainty Management for Robust Industrial Design in Aeronautics* (Springer, Berlin, 2018), pp. 203–211
17. V. Schulz, C. Schillings, Optimal aerodynamic design under uncertainty, in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Springer, Berlin, 2013), pp. 297–338
18. D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13**(4), 455–492 (1998)
19. A.I.J. Forrester, A.J. Keane, Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.* **45**(1–3), 50–79 (2009)
20. D. Maruyama, D. Liu, S. Goertz, Comparing surrogates for estimating aerodynamic uncertainties of airfoils, in *Uncertainty Management for Robust Industrial Design in Aeronautics* (Springer, Berlin, 2018), pp. 213–228
21. K. Shimoyama, S. Kawai, J.J. Alonso, Dynamic adaptive sampling based on kriging surrogate models for efficient uncertainty quantification, in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (American Institute of Aeronautics and Astronautics, Reston, 2013)
22. Z.-H. Han, M. Abu-Zurayk, S. Goertz, C. Ilic, Surrogate-based aerodynamic shape optimization of a wing-body transport aircraft configuration, in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* (Springer, Berlin, 2018), pp. 257–282
23. A. Merle, A. Stueck, A. Rempke, An adjoint-based aerodynamic shape optimization strategy for trimmed aircraft with active engines, in *35th AIAA Applied Aerodynamics Conference* (American Institute of Aeronautics and Astronautics, Reston, 2017)
24. T. Gerhold, Overview of the hybrid RANS code TAU, in *MEGAFLOW—Numerical Flow Simulation for Aircraft Design* (Springer, Berlin, 2015), pp. 81–92
25. T. Gerhold, J. Neumann, The parallel mesh deformation of the DLR TAU-code, in *Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM)* (Springer, Berlin, 2006), pp. 162–169
26. M. Meinel, G. Einarsson, The FlowSimulator framework for massively parallel CFD applications, in *PARA 2010—State of the Art in Scientific and Parallel Computing—Extended Abstract no. 44* (University of Iceland, Reykjavik, 2010)
27. R. M. Hicks, P.A. Henne, Wing design by numerical optimization. *J. Aircr.* **15**(7), 407–412 (1978)
28. T.H. Rowan, *Functional Stability Analysis of Numerical Algorithms*. Ph.D. thesis, University of Texas Austin, Austin, 1990. UMI Order No. GAX90-31702
29. J.A. Nelder, R. Mead, A simplex method for function minimization. *Comput. J.* **7**(4), 308–313 (1965)
30. Z.-H. Han, S. Goertz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerosp. Sci. Technol.* **25**(1), 177–189 (2013)

Chapter 15

In-flight Icing: Modeling, Prediction, and Uncertainty



B. Arizmendi, M. Morelli, G. Parma, M. Zocca, G. Quaranta, and A. Guardone

Abstract In-flight Icing consists of the accumulation of ice over the surfaces of flying crafts, namely aircraft and helicopters. It occurs when those fly through visible moisture, i.e., clouds, at temperatures below the freezing point. This phenomenon is undesirable because it compromises the safety and performance of the flying crafts. The physics of the phenomenon is complex, and it is still behind its full comprehension. Moreover, the characterization of the icing environments and their replication in experimental facilities is subject to large uncertainties. These might arise from phenomena like the complex physics of clouds, the accuracy of the measuring devices, or the resolution to reproduce flight and cloud properties, among others. This entails a reduction on the predictive accuracy of numerical models that seek to reproduce ice shapes and to assess the performance of ice protection systems. For these reasons, this research field could greatly benefit from the deployment of Uncertainty Quantification (UQ) techniques to account from epistemic and aleatory uncertainties in model predictions. In this chapter, an overview of the study field is presented to motivate collaborations between practitioners of UQ and researchers of the field. First, an overview of the physical phenomenon is introduced. Moreover, research methodologies are described with their identified sources of uncertainty. Next, the state-of-the-art modeling techniques are described together with their capabilities of replication of experiments. The characteristic modeling equations are presented as well. Finally, technologies for ice protection systems are deployed with the current regulation for flying in icing conditions.

Keywords Aerospace engineering · In-flight icing · Uncertainty quantification · Numerical modeling · Ice protection systems

B. Arizmendi (✉) · M. Morelli · G. Parma · M. Zocca · G. Quaranta · A. Guardone
Politecnico di Milano, Milano, Italy
e-mail: barbara.arizmendi@polimi.it; myles.morelli@polimi.it

© Springer Nature Switzerland AG 2021
M. Vasile (ed.), *Optimization Under Uncertainty with Applications to Aerospace Engineering*, https://doi.org/10.1007/978-3-030-60166-9_15

15.1 Introduction

The requirement for aircraft to be able to safely operate in diverse flight conditions has never been more prominent than it is now with increasing demand for fast and reliable transport. With this, the industry faces major dilemmas, that to this day, have yet to be resolved. For instance, aircraft and helicopters are occasionally exposed to atmospheric conditions which lead to the accumulation of ice, namely ice accretion. In-flight icing occurs when an aircraft or rotorcraft goes through a supercooled cloud at a temperature below 0 °C containing supercooled liquid water droplets. The droplets might freeze when impacting with the surface of the aircraft depending on the values of the influencing variables such as the temperature and the size of the droplets. The formation of ice on the surfaces is problematic because it alters the shape of control surfaces dropping their performance. In addition, it can cause the gathering and provision of inaccurate data to the pilot and the blockage of the mechanisms of control moving parts when accreted in those. Consequently, ice accretion presents a major threat to safety and performance of aircraft and rotorcraft which could lead to catastrophic accidents as it happened in the past.

In-flight icing has introduced a new challenge for manufacturers because they need to ensure safety and performance over different flight conditions. To this purpose, nowadays flying crafts are equipped with ice protection systems to eliminate or avoid the formation of ice in critical parts of the aircraft. The adequate design of these systems requires a holistic understanding of the icing phenomena and icing environments. Furthermore, the design must comply with existing regulation. Over the years, several research methods were utilized for understanding and predicting the process of ice accretion. Experimental tests were performed both in-flight and in wind and ice tunnels. The icing conditions are often replicated by spraying supercooled droplets into aircraft surfaces. The characterization of icing conditions is limited due to the vast amount of different icing scenarios and the resources available for these purposes. Physical models and their corresponding numerical models help understand the physics of the phenomena. These models need a thorough experimental validation in order to extrapolate the obtained conclusions into reality.

There are large uncertainties on the understanding and characterization of the ice accretion phenomenon. The replication of real icing conditions presents a major challenge due to the variability of the aforementioned conditions, to the large number of influencing parameters, to diverse icing conditions, and to the ability of the experimental facilities available to reproduce them. This would require a very large number of tests which would imply infeasible costs. In addition, flying under icing conditions exposes the flight crew and passengers to unwanted risk situations. These uncertainties lead to ambivalence on the replication of these conditions in icing wind tunnel tests and the consequent ice shapes accreted in the aircraft components. They also influence the data for experimental validation of the numerical models whose input is also subject to uncertainty. To conclude, there are many and large sources of uncertainties in the study of in-flight icing. Collaborations

and discussions within the Marie Skłodowska-Curie ITN UTOPIAE on Uncertainty Treatment and Optimization in Aerospace Engineering can enlighten the study of the ice accretion phenomenon by providing methods to identify and quantify the sources of uncertainty and to quantify the uncertainty of the model output. This can provide more robust predictions of ice shapes on aerodynamic surfaces and consequently derive more efficient and optimized designs of ice protection systems. The treatment of the uncertainty will help enhance the robustness of the obtained results reducing the needs for expensive experimental and flight tests.

The first section of this chapter presents an overview of the ice accretion phenomenon. It starts with an introduction of the icing conditions and environments, then it describes special icing interactions. The identified uncertainties will be introduced. Then, the diverse types of ice are described as well as the research methods applied and the impacts on the performance of aircraft. The second section explains the approaches for the physical and numerical modeling of the ice accretion phenomenon. The following section describes the available ice protection technologies. Next, the regulation in ice protection is described. Finally, the concluding remarks regarding uncertainties in in-flight icing are included.

15.2 In-Flight Ice Accretion

Ice accretion consists of the accumulation of ice on man-made structures when interacting with supercooled clouds, ice crystals, hoar, frost, and freezing precipitations such as snow, freezing rain, and freezing drizzle. Supercooled clouds are made of droplets whose temperature is lower than the water freezing point but still remain in the liquid phase. This is due to the lack of an external perturbation causing nucleation. These droplets can freeze following an interaction (impact) with structures causing the accretion of ice over the structures themselves. The behavior of water and ice in the ice accretion process is still subject to large uncertainties in the characterization and replication of icing conditions. Several man-made structures designed to work outdoors at high altitudes and in alpine environments are prone to experience ice accretion. Aircraft and helicopters fly at high altitudes and probably through clouds. Wind turbines, power lines, and antennas can be placed at high altitudes. The severity of ice accretion is dependent on the size of the cloud droplets, the amount of liquid water and the time of the exposure to icing conditions. In the case of aircraft, the encounter of large amounts of supercooled large droplets in clouds was found to be the most critical scenario [1]. Ice accretion causes the exposed structures to operate under different conditions than the designed ones. Intuitively, ice increases their weight causing additional loads. Furthermore, ice alters the geometry of the structures and this modifies the interaction with the surrounding airflow and the aerodynamic forces experienced by the structure, thus severely increasing drag and reducing lift. The consequences might be unexpected and they depend on the severity of the icing encounter. In any

case, those consequences are always unwanted and always imply a performance penalty.

For aircraft, which will be the focus of this chapter, the consequences of a malfunctioning go beyond economic and reliability issues. Fatal aircraft crashes caused by in-flight icing caused life losses in the past as it is evidenced in accident investigations [2–4]. Many aircraft are equipped with ice protection systems, especially those which will be allowed to fly in icing conditions. However, aircraft can be exposed to very different icing conditions and the adequate performance of the protection systems must be ensured [5]. There are large uncertainties regarding the icing conditions which lead to large uncertainties on the study of the ice accretion phenomenon. The challenge intended to be addressed by the analysis and study of ice accretion is to ensure any aircraft is protected against the most harmful possible icing condition [6]. Deep knowledge in this field will help on the design of more efficient icing systems.

This section presents an overview of the in-flight icing phenomena. It starts with an introduction to the scenarios where aircraft icing occurs. Then, the icing relevant parameters are described together with their impact on the severity of icing and the identified uncertainties. Depending on the values of these parameters, the ice accreted will present different characteristics that are described in the following section. Later, possible interactions between the droplets and the aircraft surfaces are explored. The different methods for investigating ice accretion are shown and finally the effects that the aircraft icing has on the performance and safety of the aircraft are described.

15.2.1 Icing Environment

Ice accretion can be caused by flying through supercooled clouds. But it can also be caused by the impact of snow or by freezing rain, where supercooled large droplets fall from the clouds. Furthermore, hazardous in-flight ice accretions are caused when large droplets are encountered.

15.2.1.1 Cloud Formations

Clouds are clusters of small visible droplets. Their formation requires an uplift air motion of humid air. The portion of raising air cools down and the moisture condenses forming small droplets. If the temperature of the cloud decreases and the water droplets contain no crystallization nuclei, the droplet will remain in a liquid state below the melting temperature in a metastable phase called supercooled [7]. Any perturbation such as impacts on the aircraft surface will alter this stability causing the total or partial freeze of the droplets.

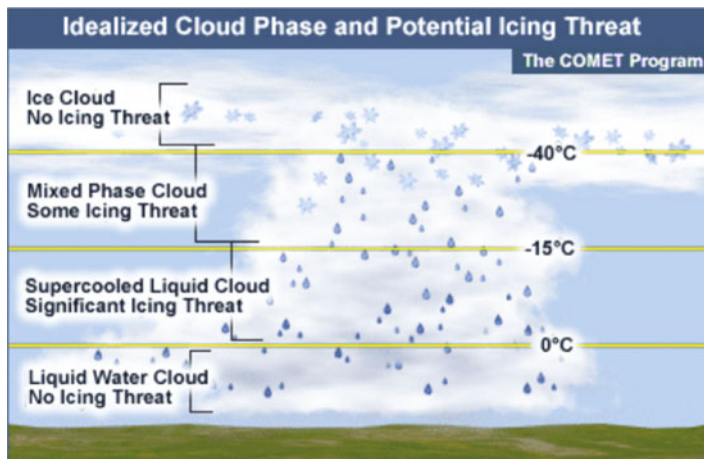


Fig. 15.1 Temperature dependent icing threats. For temperatures lower than -40°C or greater than 0°C the icing threat is minimum. Source: weather.gov

The clouds are classified according to their shape in three main blocks that are cumulus, stratus, and cirrus. Cumulus look like cotton balls and they are generated by vertical convective moist air motion. These clouds extend vertically and their bases are found at altitudes of around 2 km [8]. Cumulus can contain a high liquid water content and this potentially causes severe ice accretions when aircraft go through the supercooled ones. Stratus are stratified clouds which are generated by progressive uplifting of layers of air extended horizontally. These clouds commonly present low water content. Nevertheless, they can contain large drizzle drops [9] which poses a major hazard for aircraft safety and performance, as presented in the Sect. 15.2.6.1. Cirrus are clouds formed at very high altitudes of approximately 7 km [8]. Due to the low temperatures found, cirrus are formed by solid ice crystals. These crystals bounce back and do not accumulate on the surface of the aircraft when they impinge on it [10], as it is depicted in Fig. 15.1.

15.2.1.2 Supercooled Large Droplets

Supercooled large droplets (SLD) comprise those of a size between 50 and 500 μm according to the FAA [11]. A particular field of interest in research is the ice accreted in SLD conditions because it was claimed to be the cause of several aircraft accidents [2, 3] which were equipped with ice protection systems. SLD are also encountered in freezing drizzle and freezing rain. Freezing drizzle consists of large size droplets contained in clouds that are formed by coalescence of smaller droplets.

When the droplets impact on the aircraft surface they totally or partly freeze. Freezing rain consists of supercooled precipitating rain which also freezes when impacting with the aircraft frame. This implies that ice accretion can occur even

when flying through clouds is avoided. This kind of ice accretion might cause an increase on the surface roughness or the generation of prominent ridges or horns in the region of the leading edge or downstream [1].

Since the droplets are larger in size, inertia forces predominate to aerodynamic ones and that make the droplets follow ballistic trajectories which increase the efficiency of water collection near the leading edge [12]. In addition, the impingement point is aft the stagnation point and those can accumulate in unprotected areas. Large droplets are associated with a larger amount of latent heat. If the supplied cooling upon impact is not sufficient for complete freezing, a portion of the droplet will remain in liquid state and will run back and freeze. This may generate glaze ice ridges on the surface of the wing which might freeze in unprotected areas, and this poses a hazard for the performance and safety of the aircraft [13].

The study of the ice accretion caused by SLD presents a challenge due to the complicated physics which includes droplet shape and size changes, splashing, bouncing, and gravity effects [6]. All these respects need to be considered in the numerical modeling for accurate predictions. Furthermore, it is more likely that the droplets partially freeze because the available cooling is lower than the released latent heat of solidification. The liquid fraction runs downstream driven by the airflow and generates complex shapes of challenging prediction [14].

15.2.1.3 Ice Crystals

As the flight altitude increases and the static temperature drops, the likelihood of finding supercooled liquid droplets significantly decreases. Instead, ice crystals can be encountered. Ice crystals do not present a major threat for the aerodynamic performance of the aircraft, since the ice crystals bounce off the surface of the aircraft and they do not attach to the airframe because they are removed by aerodynamic forces. However, high densities of ice particles, which can present diverse sizes are associated with engine power losses and damages [15]. Past studies [16] reported that ice crystals melted due to relative higher temperatures attained in the compressors and that water froze in static surfaces such as the compressor second stage stator or the core of the engine. This icing type can cause obstruction of the airflow which entails working under very different conditions to the designed ones [17].

15.2.1.4 Snow

The presence of snow can also entail icing in man-made structures. Snow consists of formations of ice crystals that precipitate from the clouds. Two different types were identified in research, namely dry and wet snow. Dry snow is produced when the water from the ice crystals is in a solid state at a very low temperature. The ice crystals present very low adhesion forces among them and with surfaces. Therefore, usually there is no ice accretion from dry snow [18]. Furthermore, the crystals

might bounce off the surface and then are eventually removed by aerodynamic forces. In contrast, wet ice is formed when ice crystals fall into a higher temperature atmospheric layer and they partially melt. The range of temperatures in which wet ice can occur is between -7°C and 3°C . Adhesion forces are higher than in the case of the dry ice and therefore ice accretion is possible [19]. The formed ice consists of a porous mixture of ice, liquid water, and air that presents a low density. The impinging water can travel through the pores and freeze generating complex accretions.

Wet snow accretion in power lines has been an active field of research. Its effects include high mechanical stresses that can produce overload and failure with the potential risk of blackout or collapse [18–20]. On a different note, limited research was found in the field of aircraft snow accretion, and only very recently the topic is being addressed. Nevertheless, in order to comply with the icing regulations for certification, aircraft must be protected against snow icing envelopes which is presented in the Sect. 15.4.3.

15.2.2 Icing Relevant Parameters

Aircraft icing entails very complex physical processes which are highly dependent on many interrelated parameters. Those describe characteristics of the icing environment, the aircraft design and operation, and the interaction between those two. In addition, the duration of the icing encounter determines its severity being the longer the encounter the more severe. This section presents the environmental parameters (cloud liquid water content, ambient temperature, and droplet size), and the operational parameters (flight speed and altitude) that dictate the characteristics and severity of the icing encounter together with typical values for those parameters.

15.2.2.1 Outside Air Temperature

The main parameter influencing the accretion of ice from supercooled clouds is the steady ambient air temperature termed the outside air temperature (OAT) [12]. It is relatively easy to monitor but it implicitly only represents part of the accretion physics. Its value influences several phenomena: the convective heat transfer from the impinging droplets to the aircraft surface and the temperature of the aircraft surface. The convective heat loss over a control volume \dot{Q} can be calculated as follows:

$$\dot{Q} = h_c A (T_s - T_\infty), \quad (15.1)$$

where h_c is the convective heat transfer coefficient, A is the surface area, T_s is the surface temperature, T_∞ is the freestream temperature or Outside air temperature (OAT), and A is the surface area considered. As it is shown in Eq. (15.1), low OAT

increases the available convective cooling, which takes a larger amount of latent heat than the droplets release when they change phase. Thus, lower OAT values increase the likelihood of the droplets freezing when impacting on the surface. The requirement for the ice to accrete on the surface is that the surface temperature is maintained below 0 °C. Some ice protection systems raise the surface temperature over the freezing temperature to avoid ice accretion. Based on icing encounter measurements, Politovic [13] stated the common range of temperatures measured was between 0 °C and -25 °C with an average of -10 °C. However, very few encounters were found at temperatures below -20 °C and greater than -5 °C.

It is also important to acknowledge that the occurrence of a total temperature greater than 0 °C alone at the surface of the body does not suffice to ensure that ice accretion will not take place. For the global computation of the energy balance, the evaporative heat \dot{Q}_e for a certain control volume can be calculated as follows:

$$\dot{Q}_e = \dot{m}_e L_E, \quad (15.2)$$

where \dot{m}_e is the mass of water evaporated and L_E is the latent heat of evaporation. \dot{m}_e can be calculated as:

$$\dot{m}_e = \frac{0.7}{C_p} h_c A \frac{p_{v,s} - RH p_{v,e}}{p_e}, \quad (15.3)$$

where C_p is the specific heat at constant pressure, $p_{v,s}$ is the saturation pressure at the surface, RH is the relative humidity, $p_{v,e}$ is the saturation pressure of air, and p_e is the absolute pressure. Several droplets absorb latent heat to evaporate, thus reducing the surface temperature and allowing ice accretion even at temperatures greater than 0 °C.

The static air temperature can be measured by means of thermocouples. One of the challenges to obtain precise readings is that the probe should be sheltered from any airflow such as the main stream or the exhaust from the motors. The outside temperature can be also computed from the total air temperature subtracting the heating provided by the airspeed.

15.2.2.2 Liquid Water Content

In addition to the temperature, the liquid water content (LWC) is a critical parameter. The LWC measures the mass of liquid water that can be present within certain volume of a cloud. It is expressed as the grams of liquid water per cubic meter of air. The heat rate of freezing \dot{Q}_l , for a certain control volume over a dry surface can be quantified as:

$$\dot{Q}_l = \dot{m}_{fr} L_f, \quad (15.4)$$

where \dot{m}_{fr} is the mass of freezing water and L_f is the latent heat of fusion. The mass of freezing water is dependent on the LWC of the cloud and so does the type of ice accreted. Consequently, the larger the value of LWC, the greater the amount of latent heat which must be removed from the impinging droplets to freeze on impact. If part of the water remains in liquid state, this will run downstream by aerodynamic forces.

Higher rates of ice accretion can occur when the clouds present higher values of LWC. Typical values are in the range of 0.3–0.5 g/m³ for the 90% of the stratiform clouds and between 0.5 and 0.7 g/m³ for convective clouds [13]. The maximum LWC value the authors found was of 1.3 g/m³. It was also noted from icing encounter reports that at very low temperatures, large droplets were rarely encountered.

Hot wire probes are widely used to measure LWC in-flight. The hot wire is kept at 100 °C and the LWC is proportional to the amount of heat needed to fully evaporate the impinging droplets. The uncertainty of these measurements ranges 20–30%, increasing the larger the LWC and the MVD. This is caused by partial evaporation of large droplets or high values of LWC. Further details can be found in [21].

15.2.2.3 Airspeed

Although not strictly itself an icing parameter, the airspeed has a significant influence in determining the rate of accretion. The higher the airspeed, the greater the intercepted volume of air per unit time and hence the larger the amount of mass of water available to accumulate on the body [12]. In addition, the heat transfer coefficient h_c that determines the convective heat flux on the surface of the aircraft for laminar flow could be alternatively computed by the expression

$$h_c = \frac{0.296K}{\sqrt{\frac{\mu}{\rho}}} \sqrt{\frac{V_\infty^{2.87}}{\int_0^s V^{1.87} ds}}, \quad (15.5)$$

where V_∞ is the freestream velocity, K is the thermal conductivity of the air, μ is the freestream viscosity, ρ is the freestream density, and s is the surface distance from the stagnation point. The available convection is dependent on the airspeed, thus the transfer of latent heat to the air stream. This can influence the total or partial freeze of the impinging droplets.

Another phenomenon is the so-called aerodynamic heating. This consists of the raise in temperature of bodies exposed to a significant boundary-layer friction caused by high air speed. The kinetic energy of the fluid is transformed into heat through friction and this significantly increases the temperature of the surface to the point that it can keep a surface outside of icing conditions even when the temperature of the outside air is below freezing [22]. The contribution of the aerodynamic heating \dot{Q}_a to the energy balance can be accounted as:

$$\dot{Q}_a = h_c A \frac{V_\infty^2}{2C_p}. \quad (15.6)$$

The raise in temperature increases the value of the heat flux from the surface to the water impinging or the ice accreted and this reduces the chances of the ice to accrete. In reference [22] the authors claim that an aircraft flying at Mach number greater than 0.6 is exempt from the most severe icing conditions due to the aerodynamic heating.

15.2.2.4 Altitude

Along with the airspeed, the altitude is also not directly a relevant icing parameter. It does, however, indirectly influence many parameters which affect ice accretion rates. That is, critical combinations of icing parameters are found at certain altitudes. Most notably, the atmospheric air temperature decreases with altitude to a nearly constant value, which is called the lapse rate. Hence, for a particular range of altitudes, there is an increased potential for condensation of water droplets and their supercooling. At very large altitudes, cirrus clouds are found which are conformed by dry ice and this does not pose a major risk for the aircraft as discussed in the Sect. 15.2.1.1. The measurements of the altitude can be derived from the static pressure which can be measured by flushed-mounted static ports. These readings are generally rather precise because they indicate the exact position of the aircraft that is to be reported for control purposes.

15.2.2.5 Droplet Size

The droplet size also influences the rate and type of ice accretion. In general, the larger the droplet size, the greater the quantity of water that impacts on the aircraft over time. The droplet size can also affect the collection efficiency of drops on the airframe. Collection efficiency is defined as the ratio of the actual mass of impinging water to the maximum value that would occur if the droplets followed straight line patterns [23], further details can be found in Sect. 15.3.3.2. Droplets with a smaller diameter and mass tend to follow the stream lines. Droplets with larger diameters and mass are able to cross airflow streamlines resulting in the possibility of ice accretion further downstream when there is dominance of inertia forces. Hence, the ratio of the inertia to the aerodynamic forces determines whether the droplet impacts on a surface. An example of the comparisons between the droplet size can be shown in Fig. 15.2.

Within any single cloud, the diameter of droplets is not uniform and can differ greatly. The spectrum of droplets that exists in practice is frequently characterized by the (MVD) described in Sect. 15.2.2.2. The approximation of the spectrum

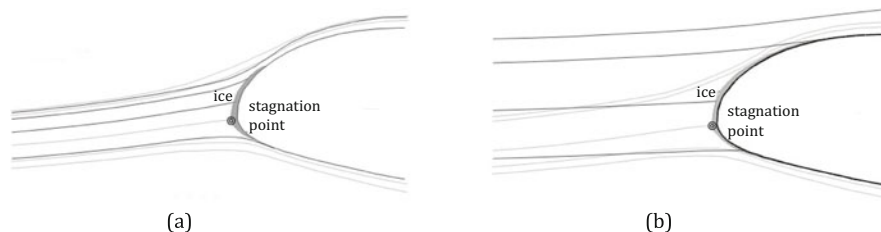


Fig. 15.2 Streamlines and droplet trajectories comparing relatively small and large droplet trajectories. The lighter lines are the streamlines and the darker lines are the trajectories. (a) Relatively small droplet size. (b) Relatively large droplet size

by means of the MVD was proven to deliver equivalent values of the collection efficiency [24]. Therefore, the expected accreted ice should be similar.

Spectra of clouds droplets can be measured by optical probes such as the Forward Scattering Spectrometer probe (FSSP-100). The probe computes the size of the droplets by means of the light that they scatter from a laser beam. The counted droplets are binned depending on their size and then the concentrations of droplets for several range of sizes are obtained. Diverse uncertainties were identified for the readings obtained such as dead time losses or coincidence of droplets. MVD values can be computed from the concentration values for the different droplet sizes. The expected uncertainty on MVD readings from the FSSP-100 lays around 12%. Further details on the probe can be found in literature [25].

15.2.3 Icing Types

Depending on when and how the supercooled droplets freeze, and the combination of values of the relevant parameters, the ice accreted will have different characteristics. This section presents the three different types of ice that can accrete, namely rime, glaze, and mixed ice types. Furthermore, the typical scenarios where those can occur and the properties of each type of ice are presented.

15.2.3.1 Rime Ice

Rime Ice is characterized by an opaque appearance and low density. This is due to air bubbles becoming trapped during the phase changing process. Rime Ice largely dominates during conditions where temperatures below -25°C exist, where supercooled water droplets instantaneously experience phase change into ice when impacting on a surface. Rime Ice is particularly prevalent at low speed, when there is low kinetic heating and for low values of LWC [13]. In these scenarios, the likelihood of encountering supercooled liquid water droplets is high. In such conditions, it is expected that ice crystals would not contribute to ice accretion. This

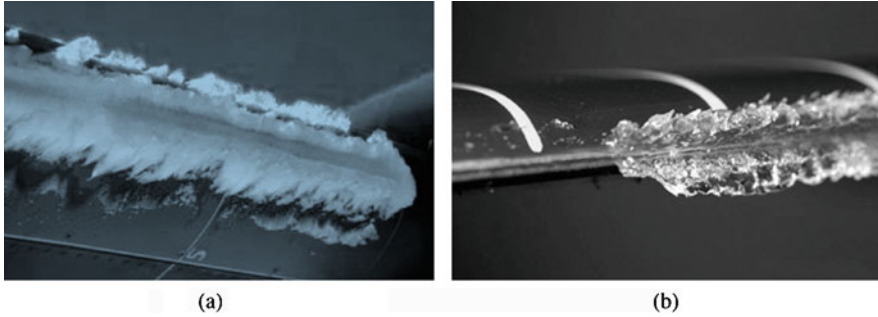


Fig. 15.3 Rime and glaze ice accretion types. Depicting the differences in their formation showing particularly the “spearhead” characterized by rime ice accretion and the upper and lower “horns” a result of glaze ice accretion. Source: NASA. **(a)** Rime. **(b)** Glaze

is due to the hard particles bouncing off the hard rime surface during impact. In the event of large ice accretions, the rime ice profile may take the form of a pointed “spearhead.” This can happen because of large exposure to icing conditions or high collection efficiency values. In Fig. 15.3(a), a typical Rime Ice formation over a wing is shown.

15.2.3.2 Glaze Ice

Glaze Ice presents a transparent look as that of the ice cubes. Glaze Ice dominates at temperatures closer to melting point, with the presence of a higher liquid water content. It is prevalent at higher speeds, due to high kinetic heating and high cloud water content. Here, water droplets first hit the surface, all the amount of latent heat cannot be transferred and consequently a portion of the water will still remain in liquid state. Then, the liquid portion flows over the wing before freezing. The resulting liquid film is usually referred to as runback water. In Glaze Ice the air bubbles are allowed to separate from the water generating a more uniform structure. As a consequence, its density is usually higher than that of rime ice. It is possible that during the brief period before the water freezes while retaining its liquid flow properties, it may accumulate aft, producing localized thickening of the ice profile into a characterized “horn” appearance on each of the upper and lower surfaces of the blade or wing. In Fig. 15.3(b), a typical glaze ice formation over a wing is shown.

15.2.3.3 Mixed Ice Conditions

Due to the variations in local velocities it is possible that both rime and glaze ice accretion are present at the same time. For example, in helicopters rime ice can accumulate at inward radial stations, while glaze ice can form further outward on the rotor.

15.2.4 Aircraft Icing Interactions

Water in solid and liquid phases interacts with the aircraft frame when it goes through icing conditions and after the icing encounter. This section aims to describe several interactions that present a complex physics, which have been studied in the literature. Surface interaction describes the dependence of the surface properties on the behavior of the droplets over it. Accordingly, ice shedding studies aim to understand the physics and effects of the detachment of ice from the surfaces. Drop impact studies intend to understand the process of collision and evolution of the water droplet shape when colliding with a surface. Finally, crystal bouncing presents the interaction of ice crystals present on clouds with the airframe.

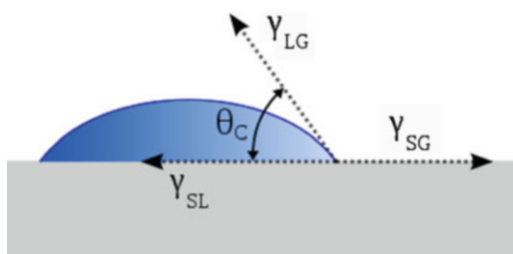
15.2.4.1 Surface Interaction

Of particular interest for ice accretion is the study of the interaction of the supercooled water liquid droplets-ice accreted with the surfaces to which they collide/ adhere to. The characteristics of the interaction of liquid droplets with a given surface depend on the wettability of the surface itself. The wettability describes the spread of a droplet when contacting a certain surface. It can be characterized by the contact angle (θ_c) which is the angle to which a sessile droplet meets the surface of the solid which is depicted in Fig. 15.4. The contact angle is unique for each liquid/solid/gas system and those can be classified as follows:

- Superhydrophilic systems if $\theta_c < 5^\circ$
- Hydrophilic systems if $5 < \theta_c < 90^\circ$
- Hydrophobic if $90 < \theta_c < 150^\circ$
- Superhydrophobic if $\theta_c > 150^\circ$

The above contact angle is defined in static conditions (static contact angle) and it generally differs from the so-called dynamic contact angle defined for droplets in relative motion with respect to the surface. The use of super-hydrophobic materials has been recently investigated experimentally to reduce the formation of ice on aircraft surfaces because the liquid droplets present low adhesion to the surfaces. Among others, Antonini et al. attempted to understand experimentally the

Fig. 15.4 Contact angle θ_c between a fluid and a hydrophilic surface. Source: Wikipedia



behavior of droplets impinging in super-hydrophobic tilted surfaces [26]. It was discovered that under certain surface angle and velocity, the droplets rebounded after impacting. Furthermore, it was found that the implementation of super-hydrophobic coatings over the surface of the wings could reduce the rate of ice accretion [27]. However, it was acknowledged that the use of these types of coatings required additional ice protection means due to the physical degradation of the surface itself. Further research from Antonini and collaborators found out that the use of super-hydrophobic coatings could decrease the energy consumption of the fully evaporative thermal anti-ice protection systems [28]. From a review of literature it is concluded that the use of coatings could support the performance of ice protection systems by decreasing the amount of icing over critical surfaces.

On a related note, the adhesion of ice to solid surfaces has also been investigated over more than 50 years which highlights the relevance of the field [29]. This research has been mainly conducted experimentally, to understand the mechanisms of ice adhesion. Over the so-called icephobic materials, the ice accreted presents low adhesion and therefore, if the layer of ice is large enough it detaches from the surface and sheds. Research intended to link “icephobia” to wettability of the surface. However, the link is not clear since some contemporaneous authors claim it is independent [30] and other authors relate the two properties [31].

15.2.4.2 Ice Shedding

Ice shedding consists of the detachment of the accreted ice on a surface of, e.g., aircraft, rotating propellers, and helicopters. The study of the trajectories of large ice particles is critical because those can impact over distant surfaces and cause damages. With regards to the ice accreted over aircraft leading edge, the shed particles can be ingested by the engines aft and cause damages. The stresses on the accreted ice layer that cause the shedding are originated by inertia forces, aerodynamic pressure forces, thermal stresses, and also flexing of the aircraft wings. The numerical studies of ice shedding conducted by Scavuzzo concluded that there is a critical value of the aircraft Mach number to which the aerodynamic forces are significant enough to cause shedding in typical aeronautic applications [32].

Ice shedding in aircraft was studied by means of numerical models, for instance, Zhang et al. developed a Finite Element Analysis (FEA) model to study ice mechanics by means of a fracture mechanics framework [33]. It was concluded that the shape of the ice strongly influences the shedding onset. In addition, Papadakis et al. performed a complete study to model the trajectories of the ice shed by means of a multi-body dynamic solver and CFD [34]. Moreover, included experimental aerodynamic characteristics of the ice particles into the numerical model. The authors acknowledged the randomness of the process. Consequently, they performed a Monte Carlo approach to quantify the probability of the icing trajectories and to find the most likely ones. The numerical modeling of the trajectories of ice particles shed can help the design of de-icing systems to ensure those particles do not impact on critical components or surfaces.

15.2.4.3 Drop Impact

The research on the behavior of the impact of drops over solid surfaces or over fluid films is a very active field. Although the first paper dates back to 1908, the process is not fully understood [35]. Drop impact is very relevant on the field of ice accretion study because it can describe in detail the interaction of the droplets with the aircraft surfaces. In addition, it is key to the development of super-hydrophobic materials or coatings that decrease the adhesion of the supercooled droplets on the surfaces of the aircraft and thus the ice accreted. The characteristics of the impact are unique for each fluid/gas/surface system [35]. Furthermore, the related parameters are the density, viscosity, diameter, surface tension of the droplet, and impact velocity of the droplet. The surface roughness and the wettability are also relevant to characterize impact on solid surfaces [36]. The governing non-dimensional parameters are the Weber number, the Reynolds number, and the Ohnesorge number. The Weber number relates the droplet inertia and surface tension,

$$We_p = \frac{\rho V^2 l}{\sigma}, \quad (15.7)$$

where l is a characteristic dimension of the model and σ is the surface tension of the droplet. The Reynolds number which describes the relationship between the inertia and viscous forces on the droplet,

$$Re_p = \frac{\rho D_p V}{\mu}, \quad (15.8)$$

where D is the diameter of the parcel. The Ohnesorge number of the particle (Oh) relates viscous forces with inertia and surface tension of the droplet and it is computed as:

$$Oh = \frac{\sqrt{We_p}}{Re_p}. \quad (15.9)$$

Experimental studies were conducted by using high resolution cameras to capture the evolution over time of the shape of the droplet when colliding with a surface. For instance, the work reported in [36], which characterized six different types of impacts that are the deposition, prompt splash, corona splash, receding breakup, partial breakup, and rebound. Numerical investigation has been conducted, for instance, Yarin and Weiss [37] who developed a mathematical model to describe the evolution of the radius of the crown generated by the impact of the drop. Furthermore, computational studies were performed to numerically model the behavior of the droplets. Brambilla and Guardone [38] used the Navier-Stokes equations with a “Volume of Fluid” approach to study oblique droplet impact on liquid films. One of the identified challenges is capturing secondary droplets in the corona splash.

15.2.4.4 Crystal Bouncing

The numerical replication of the crystal bouncing phenomenon is challenging. Ice crystals bounce back on the surfaces of the aircraft and they do not adhere to them. Ice crystal bouncing has a complicated physics due to factors such as orientation, disintegration at extremely high velocities, and the circumstances post crystal impact. Various experiments quantified the bouncing criterion for particles impacting on wet surfaces [39]. Notably Davis et al. [40] identified that on a wet surface the tangential velocity component of the impacting particle shows negligible changes, while the losses in post-impact velocities are attributed to the significant changes in the normal velocity.

15.2.5 Existing Research Methods for Ice Accretion

To date several methods are devised to help predict and understand the physics of ice accretion. These ranging from experimental tests to numerical simulations. Experimental tests which are performed either in-flight or in wind tunnels are presented in the first subsection. Then, the numerical approaches will be presented and described. Both approaches are complementary and often used in synergy to address the complex ice accretion problem.

15.2.5.1 Flight Test and Wind Tunnel Experiments

The phenomenon of in-flight ice accretion requires further efforts to be fully understood. Physical replication of icing scenarios could increase the understanding on the physics and could provide datasets for the validation of numerical models. Very complete reviews of the research methodologies for aircraft icing were developed by Kind et al. [41] and Bragg [42]. In this section we will focus on the available experimental methodologies. Depending on the location of the experiments, they are divided into ground tests and in-flight tests. Ground tests entail the use of wind or icing tunnel which vary depending on the components being tested such as fixed wings, engines, or propellers. This approach is convenient because it enables the focus on the physics of certain specific parts and requires less time and cost than in-flight testing. In addition, operating in-ground does not compromise the safety of the flight crew or the aircraft. However, this leads to uncertainties that will be presented later in the section. The main ground tests approaches described in literature [41] are:

- Dry tests—These tests are performed in a regular wind tunnel. The expected or critical ice shapes are simulated separately and then attached to the components of interest. The objective is to understand the effects of the icing on the aircraft performance and aerodynamics that is the performance degradation entailed if ice

is accreted on the aircraft. Examples of these research approach were performed by Lee [43] and Gurbacki [44].

- Icing tests—These tests are performed in icing tunnels which are equipped with spraying water system to replicate the clouds and an acclimatization system to reach temperatures below 0 °C. They can gather data from the droplet trajectories and their thermodynamic effects as well as the complex behavior of the runback water. Also the approach is used to further understand the ice shapes generated under a range of conditions and the effects of every ice shape on the aircraft performance. Furthermore, this methodology is used to test the performance of ice protection systems. This approach was taken by Olsen [45] and Whalen [46] in their investigations of the ice accretion phenomenon.
- In-Flight tests—In-flight tests provide a realistic representation of the actual icing conditions encountered. These icing scenarios can be natural or simulated and this approach was identified as the most representative on the testing of the performance of the ice protection systems. In-flight tests are also crucial for certification, since the aircraft are expected to be able to adequately perform in these conditions, and this can only be tested in-flight to some extent. However, in-flight tests are more costly than wind tunnel tests and in addition they can compromise the safety of the flight crew. Very little material has been published in this matter, since the test outcome is usually protected as manufacturer intellectual property. These tests were classified [41] in the following manner:
 - Dry tests—They are analogous to the ground dry tests where the ice shapes are replicated. In addition, these tests intend to assess the in-flight handling qualities in iced aircraft which is generally not possible to be measured in wind tunnels.
 - Water spray tests—A tanker aircraft flights in front of the tested aircraft. The tanker contains a water tank and sprays water to replicate the real icing envelopes. Also, the aircraft can have water tanks and devices to spray water near the critical regions. These tests can be used to assess the performance of the ice protection systems and the performance of engine operation under icing conditions.
 - Flight in natural icing conditions—It consists in the controlled approach to real icing conditions. It is the most precise testing procedure since it fully replicates actual icing scenarios, however, it is rather costly and might compromise the safety of the pilots (Fig. 15.5).

15.2.5.2 Uncertainties in Icing Tunnel Experiments

Icing tunnels are a type of wind tunnels where it is possible to reproduce icing conditions to study ice accretion in different components. The airflow is cooled down and then water droplets are sprayed to represent icing envelopes. Epistemic uncertainties arise from the flow and the sprayed droplets attained within the tunnel. They reflect the ability of the devices to replicate the intended icing envelopes

Fig. 15.5 Icing tunnel tests on the NASA Lewis research center. Tests of a reduced model of a commuter aircraft. Source: NASA



including airspeed, angle of attack, LWC, or MVD. Some of these uncertainties could be reduced by further testing and retrofitting the tunnel design such as the study conducted by Arrington [47] to improve the flow quality. In addition, effects of the blockage on the icing tunnel, i.e., ratio of the airfoil chord to the test-section height and the wall-wing interaction are found to increase the uncertainties of the reproduced shapes [48].

A common practice to study the uncertainties in the icing wind tunnels is the performance of repeatability tests. The objective of the tests was to assess the quality and robustness of the results. Repeatability over several runs of a test is defined as the maximum percent deviation found from the mean value of the results. In diverse research, the repeatability of icing tests ranges between 10 and 30% [49]. The repeatability of the spray clouds was found as the main source of uncertainty [50]. The development and implementation of sophisticated mathematical tools which have been presented throughout the pages of this book could help handling these large uncertainties.

15.2.5.3 State-of-the-Art Computational Ice Accretion Methods

In the past, various numerical methods were developed to simulate ice accretion on aircraft. Tran et al. [22] developed a numerical method to help aid in predicting ice modification on airfoils including the thermodynamic effects. Using a Eulerian approach for flow theory Cao et al. [51] described a numerical simulation method in which predicting the ice accretion on three-dimensional fixed wings. While researching icing models, Tsao and Rothmayer [52] investigated the formation of surface roughness on the surface of airfoils through the stability analysis of air/liquid and water/ice on substrate interfaces. Fortin et al. [53] created a new analytical model for the calculation of roughness heights and a new geometric ice

addition model based on bisection of angle between adjacent panels. Blackmore and Lozowski [54] and Blackmore et al. [55] introduced an icing model used to describe spongy atmospheric icing, and a dendritic-growth layer was assumed to correlate the microscopic ice growth with macroscopic mass/energy conservation. Kong and Lui [56] introduced a method for predicting supercooled icing in which influences of the flow velocity on ice growth were taken into account.

Currently respective prediction codes for ice accretion also exist, namely LEWICE [57], ONERA [58], FENSAP-ICE [59], and PoliMIce [60] and others. These codes are built as modelling frameworks in which aerodynamic solvers are coupled to ice formation models. In addition, they solve energy and mass balances in order to determine the amount of ice formed and its shape. Further details can be found in Sect. 15.3.

15.2.6 Ice Accretion and Performances

The accumulation of ice in-flight on aircraft surfaces modifies the interactions between the aircraft and the environment. For instance, ice accreted shapes change the shape of the wing aerodynamic profiles and this influences the performance of the aircraft. It is highlighted that ice accretion is always unwanted and it hinders its operational performance and the safety. For this reason, aircraft must be effectively protected against ice. This section will present an overview of the most important effects of ice accretion on the performance and safety of the aircraft.

15.2.6.1 Fixed-Wing Icing Environment

Ice accumulates in all the aircraft surfaces that are exposed to the environment. The study of the impact on the performance is crucial because of the number of accidents which were proven to be caused by ice accretion in both ice protected and unprotected aircraft [61]. The understanding of these effects supports the design of better systems and the elaboration of more robust certification procedures.

The accumulation of ice, regardless of the severity, modifies the shape of aerodynamic surfaces such as wings and stabilizers and therefore alters the flow and the pressure distribution [62]. In the case of the aircraft wings it leads to a dramatic increase in drag and a drop in lift. In addition, the maximum lift coefficient decreases as well as the maximum angle of attack. The stall speed is also increased. The change in shape experienced by the stabilizers, particularly the horizontal stabilizer, decreases its critical angle of attack and reduces the amount of negative lift it generates. This decreases its ability to compensate the nose-down pitching moment which can undermine the aircraft stability [5]. Aircraft control surfaces are located at the wings and stabilizers. The changes in the pressure distribution over their surface alter the hinge moment and consequently the force required to operate them. Furthermore, with the increased profile drag from ice accretion, a

corresponding increase in power consumption is expected. The limitation of these effects is crucial in the critical flight stages, which include take off, approaching, and landing. In critical missions, the limitation of thrust available can cause severe concerns and may not suffice to satisfy regulated margin of safety requirements.

Bragg [63] elaborated a high order model which physically represents the effects of ice on aircraft flight. The value of any aircraft performance, stability, or control parameter if ice has been accreted can be calculated as follows:

$$C_{(A),iced} = (1 + \eta_{ice} k'_{CA}) C_{(A)}, \quad (15.10)$$

where $C_{(A)}$ and $C_{(A),iced}$ represent any arbitrary performance, stability or control parameter to be derived under the uniced and iced aircraft. η_{ice} represents the severity of the icing encounter and k'_{CA} measures the aerodynamic effects of ice accretion on the parameter of interest, which is constant for a given aircraft and it is dependent on the presence of ice protection systems, on the geometry and configuration of the aircraft and on the specific icing conditions. The severity parameter depends on the collection efficiency, the accumulation factor, and the freezing fraction that will be presented next. The freezing fraction represents the type of ice that it is formed and it can be obtained as follows:

$$f = \frac{\text{mass of water freezing}}{\text{mass of water impinging}}. \quad (15.11)$$

Freezing fractions close to the unit represent rime ice formation, whereas values close to zero correspond to no accretion. Then, the accumulation parameter A_c is a non-dimensional parameter that corresponds to the width of ice that would accrete over a flat plate placed perpendicular to the free stream over a time t . It can be calculated as follows:

$$A_c = \frac{V_\infty \text{LWC } t}{\rho_{ice} c}, \quad (15.12)$$

where t represents the time and c represents the chord length of the airfoil and LWC accounts for the liquid water content. The collection efficiency β will be explained in more detail in the Sect. 15.3.3.2. It can be calculated as:

$$\beta = \frac{\text{Mass of water droplets impinging}}{\text{Mass of water in the body projected area}}. \quad (15.13)$$

The general form for the increase in drag ΔC_D has the following general form:

$$\Delta C_D = Z_1 A_c \beta g(f), \quad (15.14)$$

where Z_1 is a constant and the function g decreases linearly as f differs from 0.2, where it presents a peak. Since the curve is linearly dependent on $A_c \beta$ then the performance drop will increase over time if no ice protection system is

deployed. The authors concluded that further considerations need to be included into the model. Moreover, when ice accumulates in probes and in sensors, the readings provided can be altered and this can provide misleading information about the current flying situation. For instance, if ice accumulates in the outside air temperature sensor, the reading will provide the temperature of the ice accreted rather than the actual air temperature. Some other sensors are not designed to work in icing conditions and the reading can be misleading and hinder the assessment of the operating condition [11].

15.2.6.2 Rotating-Wing Icing Environment

Rotorcraft flight in icing environments is indeed a notoriously challenging phenomenon in which the formation of the ice is inherently unpredictable. This, coupled with the complex multi-body rotorcraft aerodynamics, makes it infamously difficult to prognosticate, not to mention the extent of time required to do so. If such a scenario occurs, the presence of ice on the blades of the main rotor can lead to severely damaging consequences to the helicopters' performance capabilities, becoming a serious threat to flight safety [64] and the cause of several aircraft icing accidents [4]. It can prompt drastic alterations to the geometry and increase the surface roughness thus, resulting in the increase of drag, reduction of lift and premature onset stall. These aerodynamic changes invariably have implications on the helicopter stability, flight condition, power and torque characteristics, and component loading [65, 66]. The buildup of ice on the rotor blades can also alter the trim conditions due to disturbing the blades balanced weight as well as modifying the inertia and aero-elastic properties of the blades themselves [60].

15.3 Modeling Ice Accretion

This section will discuss the various numerical methods used to compute an in-flight ice accretion simulation described in Sect. 15.2. It will briefly describe the development of the methods in order to obtain an accurate solution. An in-flight ice accretion computation is conventionally performed using a multi-stage process, including a feedback loop to iteratively update the iced geometry as shown [67]:

- **Flow Solver:** Used to determine the flow field around the aerodynamic body.
- **Droplet Solver:** Required for computing the water droplet trajectories for the local and global collection efficiency.
- **Ice Accretion Solver:** Necessary to predict the final ice shape around the aerodynamic body.
- **Mesh Morphing:** Implemented for updating the newly accreted iced geometry into the flow solver.

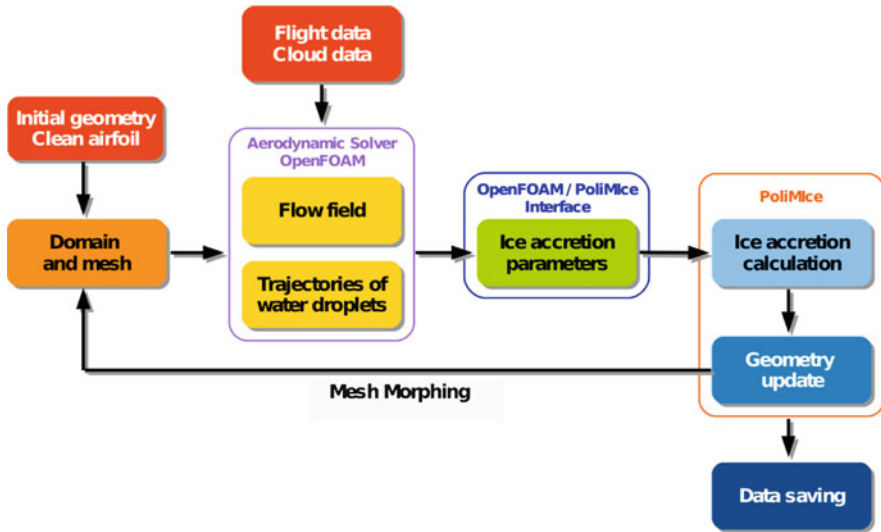


Fig. 15.6 Flowchart of the PoliMIce solver simulation depicting the process of an ice accretion simulation taken from Gori et al. [60]

Gori et al. [60] provide an exemplary framework for in-flight icing simulations displayed in Fig. 15.6. Here, the initial clean airfoil geometry is required to be meshed. This geometry is then passed on to the aerodynamic solver where the flow field is computed around the imported geometry. This work utilized the multi-physics capabilities of both SU2 [68] and OpenFOAM [69] for this purpose. These required data determining the flight and cloud information. The flow field information is then passed onto the particle tracking software PoliDrop which computes the trajectories of the water droplets. This allows the collection efficiency to be calculated by counting the number of droplets collected in each cell. An interface then links the collection efficiency data to the ice accretion software. The ice accretion software PoliMIce [60] was then able to compute the new ice thickness from the amount of liquid in each cell. The airfoil geometry is then updated by a mesh morphing algorithm and fed back to the start of the process which repeats itself for the extended period of icing.

15.3.1 Flow Field Determination

A precise modeling of the flow field around and on the aerodynamic surface and growing ice shape serves two important purposes. The incoming flow field affects the location of the impinging water droplets due to its influence on the particles trajectories, while the viscous flow characteristics on the surface are imperative to understand the convective heat transfer fluxes, which have a major impact on ice

accretion rates. Physical phenomenon such as the transition of the boundary layer as well as its detachment have an effect of the convective heat transfer fluxes. The study of these factors should be carefully considered [6]. However, determining the most accurate method for modeling the aerodynamic flow field in icing conditions is under discussion. Ideally, the prerequisite for the solver should be that it is capable of handling complex aircraft and ice geometries and complicated flows. Severe ice accretion scenarios that present complex and large ice formations poses a big challenge for the development of computational models. An example are the ice horns such as the ones described in Sect. 15.2.3, pose large challenges for many computational methods. The abnormally large and inherently irregular increase in roughness also something that methods should be able to account for.

15.3.1.1 Flow Solver

Commonplace aerodynamic flow field prediction methods implement various ways of modeling. Examples of these techniques are linear panel methods with integral boundaries [70], interactive boundary layer representation [71] and Reynolds-Averaged Navier-Stokes simulations (RANS) [72]. These methods present different levels of complexity and accuracy. Precise predictions of flow features for a range of geometries and flow conditions are a mandatory goal for flow solvers. For icing purposes, the flow solver is required to reliably predict separation onsets and progressions in highly curved surfaces [6]. With complex ice structures, such as horns, this becomes very challenging. RANS simulations are the most widespread modeling approach for fast and generally reliable predictions of flow field. Reynolds-Averaged Navier-Stokes (RANS) equations require a turbulence model. This decomposition of the Navier-Stokes equations into the RANS equations introduces a set of unknowns called Reynolds Stresses, which are functions of the velocity fluctuations and utilize a turbulence model for closure of the system [73, 74]. The time-average Navier-Stokes equations of motion for a stationary, incompressible fluid can be written as:

$$\rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j} \left[-\bar{p} \delta_{ij} + \mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \overline{\rho u'_i u'_j} \right], \quad (15.15)$$

where \bar{u} is the average velocity, u' are the velocity fluctuation values, ρ is the density, \bar{f}_i are the surface forces, μ is the dynamic viscosity. Every turbulent flow is unsteady. As the time-averaging of the RANS equations is based on a stationary stochastic process, parameters such as mean and variance of the flow properties change over time, this means that if the flow has a large scale periodicity such as vortex shedding RANS cannot be used. Unsteady simulations like that of helicopters during forward flight must hence use unsteady RANS computations to solve the flow field while maintaining the use of turbulence models for closure.

Each RANS simulation requires the correct turbulence model dependent upon the properties and behavior of the flow. Ice accretion is highly sensitive to the flow field thus, modeling it accurately is crucial. Moreover, with different types of ice formation comes different flow characteristics and such is the case that one type of turbulence model may not be appropriate for all icing scenarios. Some models such as the Spalart-Allmaras turbulence model maybe a good compromise between stability and accuracy, however, it may also over predict heat fluxes if assumptions such as fully turbulent flow are made. Hence, understanding the flow is essential to apprehend which turbulence model is best suited for accurately simulating ice accretion. Dunn et al. [75] investigated large droplet ice shapes on airfoil aerodynamics with the one equation Spalart-Allmaras turbulence model and found that lift and drag predictions were in good agreement with the experimental results of Lee et al. [76] until the point where the flow was fully separated. The results worsened with increasing incidence and were ascribed to the turbulence models inadequately in predicting the amount of entrainment in the shear layer. A further study from Marques et al. [77] on the same experimental data from Lee et al. [76] later found the discrepancies in the Spalart-Allmaras and the $k - \omega$ turbulence models at high angles of incidence. It found that generally the $k - \omega$ turbulence model under predicted the suction peak downstream of the ice shape and both models demonstrate difficulties in capturing the correct suction levels upstream of the ice shape.

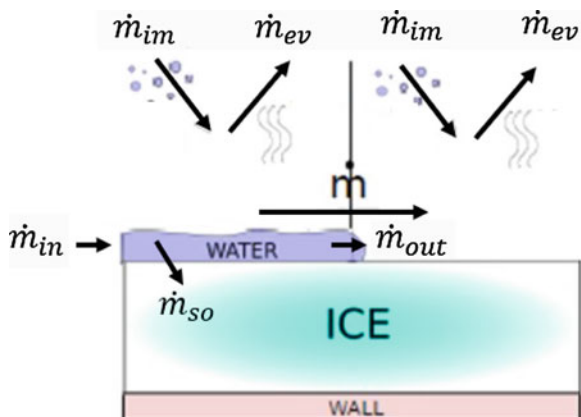
15.3.2 Governing Equations for Multiphase Flows

Modeling the mass, momentum, and energy balance properties of the incoming supercooled water droplets on a surface will now be discussed taking into consideration the conservation of both mass and momentum. Aforesaid, the incoming liquid may not instantaneously freeze upon impact but run downstream until freezing in certain conditions, known as glaze ice. This process is strongly related to the mass and heat transfer, surface roughness, skin friction, etc. over cells. The freezing fraction over a control volume can be computed applying the first law of thermodynamics with knowledge of the mass flux and a heat balance. This, together with the droplet impingement rates are required to estimate the ice thickness.

15.3.2.1 Mass Balance

The ambient temperature determines the ice type. At low temperature Rime Ice conditions dominate and the surface is effectively dry, as the water droplets freeze on impact. At higher temperatures, however, Glaze is the primary source of icing. At Glaze conditions, there are mass and energy flows on the surface of the airfoil that should be accounted for. Consequently, the motion of the droplets on impact as runback water is important. An illustration of a control volume impacted by

Fig. 15.7 Schematic illustration of mass balance for a control volume

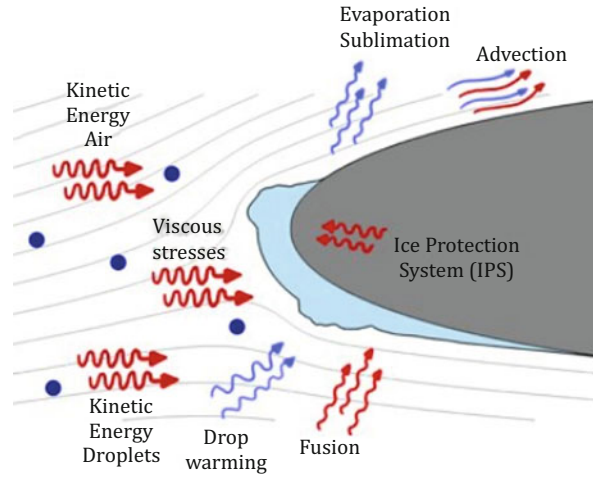


incoming water droplets is shown in Fig. 15.7, where the mass flow rate of water running from one cell to another is allowed. This is typically computed from one glaze cell to another, however, the modeling of mass transfer from a glaze to rime cells has also since been established [78].

15.3.2.2 Energy Balance

Initially, it was assumed that once the droplet impingement characteristics were established, the actual ice accretion process was governed almost exclusively by the convective heat transfer at the surface [6]. This early formulation for ice accretion was developed by Tribus et al. [79] in the late 1940s, before being developed further. Now it is apparent that the energy balance of the system has to incorporate all the possible heat fluxes, where an illustration of this is shown in Fig. 15.8. In terms of aerospace icing applications the heat flux into the system includes: heat added due to the latent heat of fusion, kinetic heating by the water droplets and viscous or kinetic air heating. Conversely, the heat flux out of the system includes: heat lost due to convection, evaporation, or sublimation and warming of the droplets. These have to be calculated for each control volume. The surface boundary layer has a strong influence on the heat transfer rates, for example, depending if the flow is laminar, transitional, turbulent, or separated [6]. Subsequently, these flow regimes are characterizations which strongly dependent upon the Reynolds number, surface and importantly ice geometry, surface roughness, free stream turbulence and in-flight pressure, and velocity conditions. Contributing to the concealed nature of this complex phenomenon, the majority of tests are conducted with small scaled models at relatively low Reynold numbers in wind tunnels. Even 2-D test measurements have documented minimal increases in Reynolds number result in significant increases in heat transfer rates over and just downstream of ice roughness [80], where the heat fluxes here were primarily described by Messinger [81] in a

Fig. 15.8 Illustration of the heat exchange terms involved in Messinger model



model for solving the Stefan condition, which is now the basis of many ice accretion solvers and will be later described during Sect. 15.3.4.

15.3.3 Droplet Solver

Computing the droplet impingement is the second step of an ice simulation process. Once the flow solution is known, a droplet trajectory calculation is required to determine the total collection efficiency, local collection efficiency, and impingement limits of the water droplets impacting on the aerodynamic body. The process for simulating ice accretion occurs when aircraft penetrate a cloud containing supercooled water droplets, freezing drizzle, or freezing rain. The fundamental mathematical investigation of water droplet trajectories was developed by Langmuir and Blodgett [82]. Since then, however, there have been various frameworks implemented with a number of alterations such as Bourgalet et al. [83] Eulerian approach to supercooled droplets impingement calculations. The trajectory of a droplet is ascertained by integrating a differential equation representing the force balance on the droplet containing the inertial, drag, buoyancy, and gravitational forces which are tracked from the incoming free stream until either impacting with or bypassing the surface. Essentially there now remain two major difference in paradigms when considering particle tracking being, the Eulerian and Lagrangian approaches which will later be discussed in this section. There are also a number of schemes available for the calculation of the collection efficiency from predetermined particle tracks. All methods relate initial cross-sectional area of the droplet stream tube to the area at impact on the surface. For the method described by Gent [84],

where two-dimensional flow conditions are modeled, the process evaluates two additional trajectories, each a small displacement either side of the trajectory. This is the method used by LEWICE, FENSAP, and PoliMIce.

15.3.3.1 Eulerian and Lagrangian Specifications for Particle Tracking

Although several codes exist for simulating ice accretion, all have been developed using different methodologies for solving the complex issue of icing, each with their own strengths and weaknesses. The majority have either used Eulerian or Lagrangian particle tracking frameworks to model the physics of ice accretion. FENSAP-ICE solves the particle governing equations using an Eulerian approach [59], meanwhile PoliMIce [60] and ONERA [58] use a Lagrangian particle tracking solver to compute the droplet trajectories. In classical field theory the Lagrangian specification of the field is a way of looking at fluid motion where the observer follows an individual fluid parcel as it moves through space and time [85]. The position of particles through time gives the path line of these particles. These fluid particles have mass, momentum, internal energy, and other properties. Mathematical laws can then be written for each fluid particle. The Eulerian paradigm of the flow field is a way of looking at fluid motion that focuses on specific locations in the space through which the fluid flows as time passes [85]. In the Eulerian specification of a field, it is delineated as a function of position, x , and time, t . For instance, the flow velocity can be characterized by a function,

$$u(x, t). \quad (15.16)$$

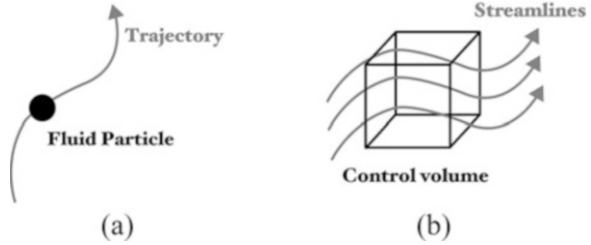
On the contrary, with the Lagrangian paradigm, individual fluid parcels are followed through time. The fluid parcels are denoted by a time-independent vector field, x_0 . More often than not x_0 is described as being the center of mass of the parcels at some initial time, t_0 to account for the possible changes of shape over time. Hence, the center of mass is a good parametrization of the flow velocity, u of the parcel. The flow given by a Lagrangian specification is described by the following function. This gives the position of the parcel labeled x_0 at time t .

$$X(x_0, t). \quad (15.17)$$

Both Eulerian and Lagrangian flow specifications are related as each side describes the velocity of the parcel denoted x_0 at time t .

$$u(X(x_0, t), t) = \frac{\partial X}{\partial t}(x_0, t). \quad (15.18)$$

Fig. 15.9 Lagrangian and Eulerian specification of a flow field. (a) Lagrangian. (b) Eulerian



The Lagrangian and Eulerian specifications of the kinematics and dynamics of the flow field are associated by the material derivative. In relation to acceleration it can be expressed as,

$$\underbrace{\frac{D\mathbf{v}}{Dt}}_{\text{Lagrangian acceleration}} = \underbrace{\frac{\partial\mathbf{v}}{\partial t}}_{\text{Eulerian acceleration}} + \underbrace{\mathbf{v} \cdot \nabla\mathbf{v}}_{\text{Material derivative}} \tag{15.19}$$

The Eulerian method accounts for the effects of diffusion and is better suited for problems that involve interfaces such as shocks and rarefaction fans. Albeit the Lagrangian formulation is highly able to be parallelized since all particles are advected independently and is easier to develop higher-order methods. A schematic view of these two approaches is depicted in Fig. 15.9.

15.3.3.2 Collection Efficiency

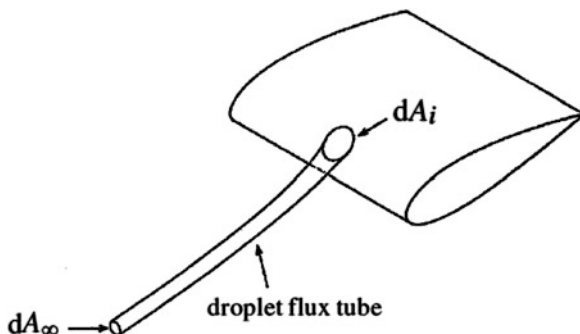
The collection efficiency is ratio of mass flux of droplets impinging on the surface to the mass flux at the free stream. A schematic view of the collection efficiency is depicted in Fig. 15.10. The collection efficiency is vitally important for understanding the ice accretion rates. In 3D the collection efficiency β can be expressed as,

$$\beta = \frac{dA_\infty}{dA_i}, \tag{15.20}$$

where dA_∞ is the symbolic form of the free stream area of the droplet flux tube and dA_i is the notation for the impacted area on the surface of the droplet flux tube.

Values of collection efficiency can range between 0 and 0.8, where the lower limit represent a clean surface and the upper limit represent high rates of ice accretion around the stagnation points. Whether a droplet does or does not impact the surface depends on the ratio of the inertia to the aerodynamic forces on the droplet. The collection efficiency is influenced by many parameters, however, small streamlined geometrical designs such as the short chord length and small leading edge radius

Fig. 15.10 3D Schematic defining the total and local collection efficiency [12]



induce high rates of ice accretion and are to be avoided if possible [12]. This, however, is not always possible, for instance, helicopter rotor blades which require a relatively short chord length making them particularly susceptible to icing.

15.3.4 Unsteady Ice Accretion

Calculating ice accretion is the final stage of the simulation process. Various complexities of ice accretion models will be discussed, mentioning how they have been developed beginning with their early stages through to the current methods used today. Their influencing parameters such as the outside air temperature, altitude, crystal bouncing, droplet size, liquid water content and rime and glaze icing types will also be discussed with how they affect the severity of icing.

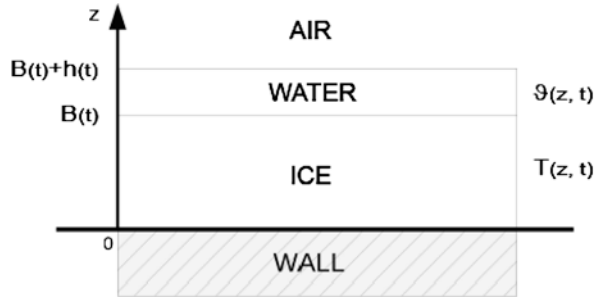
15.3.4.1 Development of Ice Accretion Models

Likewise with the different frameworks, different complexities of ice accretion models exist. The first mathematical formulation of the liquid water-ice two-phase problem was given by J. Stefan in 1889 describing the phase changes of physical systems. This was later relevant in aeronautical applications with Messinger's proposed formulation of Stefan's problem in 1953 [81]. More recently, Meyers developed a formulation which better accounts for the two different mechanisms associated with rime and glaze ice formation in 2001 [86].

15.3.4.2 The Stefan Problem

The Stefan problem is a set of four partial differential equations describing the evolution of a single-component two-phase system during a phase change. Its

Fig. 15.11 Elementary cell for the discrete ice problem. The cell reference system is also depicted. Over each elementary cell composing the entire surface the Stefan problem is solved



complete solution gives the temperature distribution within the solid (T) and the liquid layers (ϑ), the height of the ice layer (h), and the thickness of the water layer (B). This kind of problem belongs to the family of the so-called *moving-boundary problems* as the position of the solid-liquid interface is unknown and depends on the time and on the solution itself. Considering the reference system in Fig. 15.11, one-dimensional Stefan problem can then be expressed as,

$$\left\{ \begin{array}{l} \frac{\partial T}{\partial t} = \frac{K_s}{\rho_s C_s} \frac{\partial^2 T}{\partial z^2} \\ \frac{\partial \vartheta}{\partial t} = \frac{K_l}{\rho_l C_l} \frac{\partial^2 \vartheta}{\partial z^2} \\ \rho_s \frac{\partial B}{\partial t} + \rho_l \frac{\partial h}{\partial t} = \dot{m}_{in} - \dot{m}_{out} \\ \rho_s L_F \frac{\partial B}{\partial t} = K_s \frac{\partial T}{\partial z} \Big|_{B(t)^-} - K_l \frac{\partial \vartheta}{\partial z} \Big|_{B(t)^+} \end{array} \right. , \quad (15.21)$$

where ρ corresponds to the density, K is the thermal conductivity, and C is the specific heat. Moreover the subscripted index s and l stands for the *solid* and the *liquid* phase and the z coordinate is aligned along the normal of the surface. \dot{m}_{in} and \dot{m}_{out} are the runback water mass fluxes to the control volume.

The first and second equations described in Eq. 15.21 represent the heat diffusion within the solid and the liquid phase, respectively. The third equation explains the continuity relationship and enforces the mass conservation law. While the fourth term is the so-called Stefan condition and it is an energy balance relating all the heat fluxes involved in the phase change at the solid-liquid interface. It ensures that the latent heat caused by the phase change is equal to the net flux of heat from and towards the upper and the lower layers.

15.3.4.3 Messinger Model

Messinger model [81], proposed in 1953, suggested an analysis of the equilibrium temperature reached by an unheated surface in several icing conditions. It was based on the energy balance between the water (or ice) layer and the atmosphere surrounding the surface and where it solves only the Stefan condition in Eq. 15.21 which can be rewritten as,

$$\dot{Q}_l = \dot{Q}_c + \dot{Q}_e + \dot{Q}_d - \dot{Q}_k - \dot{Q}_a. \quad (15.22)$$

The heat fluxes involved in the phase change process, \dot{Q}_l , in Eq. 15.22 are: the heat entering related to air friction, \dot{Q}_a , the heat entering related to droplet kinetic energy, \dot{Q}_k , the heat exiting related to convection, \dot{Q}_c , the heat exiting either from evaporation (glaze regime) or from sublimation (rime ice), \dot{Q}_e , and the heat exiting related to the droplet latent heat, \dot{Q}_d .

15.3.4.4 Myers Model

Myers model is an extension of the original Messinger model previous outlined, where it takes into account the conduction of the heat in the ice layer. The hypothesis introduced by Myers can now be simplified under the following assumptions: The physical properties of ice and water do not depend on the temperature; the substrate is at constant temperature, usually assumed to be equal to the air temperature for aerodynamic purposes; the phase change from the water to ice occurs at a specified fixed temperature, assumed to be the freezing temperature; droplets are in thermal equilibrium with the surrounding air, so their temperature is supposed to be equal to the air temperature; the temperature profile in both the ice and water layers can be approximated as a linear function of the distance from the substrate and in aerospace applications the water layer is usually assumed to be isothermal due to the very small thickness. The simplified Stefan problem can thus be written with the following assumptions as,

$$\left\{ \begin{array}{l} \frac{\partial T}{\partial t} = \frac{K_s}{\rho_s C_s} \frac{\partial^2 T}{\partial z^2} \\ \frac{\partial \vartheta}{\partial t} = \frac{K_l}{\rho_l C_l} \frac{\partial^2 \vartheta}{\partial z^2} \\ \rho_s \frac{\partial B}{\partial t} + \rho_l \frac{\partial h}{\partial t} = \beta L W C V_\infty \\ \rho L_F \frac{\partial B}{\partial t} = K_s \frac{\partial T}{\partial z} \Big|_{B(t)^-} - K_l \frac{\partial \vartheta}{\partial z} \Big|_{B(t)^+} \end{array} \right. \quad (15.23)$$

In his formulation, Myers introduced the so-called *rime limit thickness* B_g as a criterion for the selection of the proper accretion law, thus allowing for a smooth transition between the rime and the glaze regimes.

The heat flux can be observed from Fourier's law as the thermal conductivity times the temperature, where its derivative at the phase changing interface under a steady temperature profile in the ice layer can be approximated as,

$$\frac{\partial^2 T}{\partial z^2} = 0. \quad (15.24)$$

Confined to the glaze regime, this is associated with the subsequent linear temperature profile within the ice layer,

$$T(z) = \frac{T_{freezing} - T_{wall}}{B} z + T_{wall} \quad (15.25)$$

thus leading to,

$$\dot{Q}_{down} = K_i \frac{\partial T}{\partial z} = K_i \frac{T_{freezing} - T_{wall}}{B}. \quad (15.26)$$

The ice accretion law for the rime regime can therefore be wrote as,

$$\text{Rime : } \quad \frac{\partial B}{\partial t} = \frac{\beta L W C V_{\infty}}{\rho_{ri}}. \quad (15.27)$$

The ice accretion law for the glaze regime mean while reads,

$$\text{Glaze : } \quad \frac{\partial B}{\partial t} = \frac{1}{\rho_{gi} L_F} (\dot{Q}_{down} + \dot{Q}_{up}), \quad (15.28)$$

where \dot{Q}_{down} and \dot{Q}_{up} are the heat fluxes exchanged between the phase changing interface and ice and water or air, respectively. The rime ice limit thickness describing the condition at which the glaze regime can first appear, and hence the first instant at which water begins to accumulate on the surface can now be defined. It is calculated using the Stefan condition and the mass conservation law in which the water height is set to zero and is given by,

$$\text{Rime limit thickness : } \quad B_g = \frac{A K_i (T_{freezing} - T_{wall})}{A L_F \beta L W C V_{\infty} - \dot{Q}_{up}}. \quad (15.29)$$

In this model of Myers, the ice accretion laws can be governed by:

- $B < B_g$ or $B_g < 0$: the *rime* ice accretion law is used,
- $B > B_g$: the *glaze* ice accretion law is used.

15.3.4.5 An Improved Myers Model

A new modified Myers model derived from the Stefan problem introduced by Parma et al. [78] explicitly accounts for the mass flux related to sublimation. Here, an improved method of describing the liquid film flow above the ice surface is introduced; permitting not only water flowing from one glaze cell into another adjacent glaze cell but also mass transfer from a glaze to a rime cell. This ensures to maintain the mass transfer criterion. To model this mass transfer an additional term is included for rime ice accretion.

Finally, the third modification concerns the description of the heat diffusion problem through the ice phase in the glaze regime. It introduces a modification in the temperature distribution within the ice layer which better respects the hypothesis of the high thermal conductivity of the wall. The linear temperature profile is replaced by an assigned parabolic shape function, namely $T(z) = a\sqrt{z} + b$, where a and b are two coefficients defined by the boundary conditions $T(0) = T_{wall}$ and $T(B) = T_{freezing}$ to obtain,

$$T(z) = T_{wall} + \frac{T_{freezing} - T_{wall}}{\sqrt{B}} \sqrt{z}. \quad (15.30)$$

The new modified model for the ice accretion law for the rime regime hence reads as,

$$\text{Rime : } \quad \frac{\partial B}{\partial t} = \left[\frac{\dot{m}_d + \dot{m}_{in} - \dot{m}_s}{A\rho_{ri}} \right], \quad (15.31)$$

where \dot{m}_d accounts for the mass rate of impinging droplets, \dot{m}_{in} is the runback water mass rate coming into a control volume, and \dot{m}_s is the mass rate lost due sublimation. A modified ice accretion law for the glaze regime is defined as,

$$\text{Glaze : } \quad \frac{\partial B}{\partial t} = \frac{1}{\rho_g L_F} \left[K_i \frac{(T_{freezing} - T_{wall})}{2B} + \frac{(\dot{Q}_c + \dot{Q}_e + \dot{Q}_d - \dot{Q}_k - \dot{Q}_a)}{A} \right], \quad (15.32)$$

where the limiting thickness can now be defined as,

Rime limit thickness :

$$\frac{A K_i (T_{freezing} - T_{wall})}{2 \left[L_F (\beta \text{LWC } V_\infty A - \dot{Q}_s \text{Ls}^{-1}) - (\dot{Q}_c + \dot{Q}_s + \dot{Q}_d - \dot{Q}_k - \dot{Q}_a) \right]}. \quad (15.33)$$

15.3.4.6 An Unsteady Ice Accretion Model

A model based on the exact solution of the Stefan problem accounts for the unsteady glaze ice accretion which closely follows the Myers formulation. As previously discussed in aeronautical working conditions the water film is very thin, thus justifying the assumption of infinite conduction through the water. The temperature of the liquid film can then be assumed as approximately constant and equal to the freezing temperature of water. Dirichlet boundary conditions are then imposed on the supposedly infinitesimally thin water layer resulting in the water layer being discarded from the Stefan problem which reduced to,

$$\left\{ \begin{array}{l} \frac{\partial T}{\partial t} = \frac{K_s}{\rho_s C_s} \frac{\partial^2 T}{\partial z^2} \\ \rho_s \frac{\partial B}{\partial t} + \rho_l \frac{\partial h}{\partial t} = \dot{m}_{in} - \dot{m}_{out} \\ \rho L_F \frac{\partial B}{\partial t} = K_s \frac{\partial T}{\partial z} \Big|_{B(t)^-} - K_l \frac{\partial \vartheta}{\partial z} \Big|_{B(t)^+} \end{array} \right. \quad (15.34)$$

An unsteady exact solution of this Stefan problem then uses a similarity approach in order to determine the temperature profile in the ice layer and the glaze ice accretion rate. A similarity approach requires that the boundary conditions which, in this case, the temperature of the wall and at the ice-water interface are specified.

This yields the exact glaze ice temperature profile,

$$T(z, t) = T_{wall} + (T_{freezing} - T_{wall}) \frac{\operatorname{erf}\left(\frac{z}{2\sqrt{\alpha_i t}}\right)}{\operatorname{erf}(\lambda)}, \quad (15.35)$$

where λ and α are reduction variables. The solution here is not well defined as it depends on the interface position $B(t)$, which is still unknown. This is as the parameter $B(t)$ appears as a function of the parameter λ . In order to close the problem the Stefan condition is applied and a procedure to calculate the derivatives of the temperature and of the position of the ice-water interface is formulated. The derivatives of the Stefan problem are hence required to be calculated and the Stefan condition is recalled for convenience,

$$\rho L_F \frac{\partial B}{\partial t} = K_s \frac{\partial T}{\partial z} \Big|_{B(t)^-} - K_l \frac{\partial \vartheta}{\partial z} \Big|_{B(t)^+}. \quad (15.36)$$

The calculation of the temperature gradient in the water layer at the interface requires initially to analyze the thin film approximation as previously discussed. The final form of the temperature gradient at the water-air interface is,

$$-\frac{\partial \vartheta}{\partial z} \Big|_{B(t)^+} = \frac{\bar{H}_1(T_{freezing} - T_\infty) + \bar{H}_2(T_{freezing} - T_{local}) - (\dot{Q}_{aw} + \dot{Q}_k)}{K_w} \quad (15.37)$$

$$= \frac{\dot{Q}_{up}^{(n)}}{K_w}, \quad (15.38)$$

where $\bar{H}_1 = (\bar{H}_{aw} + \chi e_0)$, being H_{aw} is the convective heat flux and χe_0 are a proportionality factor for evaporation, and $\bar{H}_2 = \beta \text{LWC} V_\infty C_w$. Alternatively from Myers formulation, the heat fluxes expressing the heat exchanged by convection and by evaporation at the water-air interface, \dot{Q}_{aw} , include the local temperature, T_{local} , of the airflow outside the boundary layer instead of the free stream temperature.

The calculation of the temperature gradient in the ice layer at the interface is evaluated at $z = B(t)^-$, leading to the temperature gradient at the ice-water interface which is given by,

$$\frac{\partial T}{\partial z} \Big|_{B(t)^-} = \frac{(T_{freezing} - T_{wall}) \exp(-\lambda^2)}{\text{erf}(\lambda) \sqrt{\pi \alpha_i t}}. \quad (15.39)$$

The calculation of the ice accretion rate from the exact temperature solution requires an iterative procedure such as, the Newton-Raphson method for it to be solved. This is as the ice accretion rate is a function of λ . For time $t = t_n$ it takes the form,

$$\text{Glaze : } \quad \frac{\partial B}{\partial t} \Big|^{(n)} = \frac{1}{\rho_{iglaze} L_F} \left(K_i \frac{(T_{freezing} - T_{wall}) \exp(-\lambda^{(n)2})}{\text{erf}(\lambda^{(n)}) \sqrt{\pi \alpha_i t^{(n)}}} + \dot{Q}_{up} \right). \quad (15.40)$$

15.3.5 Mesh Morphing

It is recognized that ice accretion is an inherently unsteady phenomenon. Multi-shot approaches have hence been used to observe ice shape features more accurately over heuristically prescribed time intervals. During these time intervals a feedback loop is incorporated consisting of a mesh morphing process which iteratively updates the newly iced geometry and passes this information back into the flow solver for the following time step calculation. Mesh morphing uses complicated algorithms to calculate the deformed geometries and these may depend whether the components are stationary or moving. The different algorithms that may be favored for each will now be described:

1. **Stationary components:** May use algorithms to *deform* and *move* the mesh at each time step. Since the deformations caused to the surface geometry from icing

are slight robust and expensive algorithms are not required. More cost-effective methods can hence be used like those based on the Laplacian operator, such as the spring or elastic methods [87].

2. **Moving components:** May use algorithms to continuously *stitch* the mesh. Here two grid domains are considered: The first being fixed around the stationary object, for example, the fuselage and the second rotating with the moving components, for example, the blade. The domains are then separated by a small gap in space which is constantly being re-meshed or stitched with tetrahedral elements. Stitching algorithms may be based on simplified advancing front techniques [88].

Mesh deformation algorithms such as these, however, usually require solving a system of equations which can become computationally intensive for complex three-dimensional applications. For aircraft icing simulations this can become problematic. PoliMIce hence implements an explicit mesh motion algorithm based on Inverse Distance Weighting interpolation [89], which does not lead to solving a system of equations. This results in a much faster mesh motion algorithm which is much easier to implement. An example of this Inverse Distance Weighting interpolation implemented by PoliMIce can be shown in Fig. 15.12. Numerically this can be described by the interpolation function $u(x)$ where u is the interpolated value at a given point x based on samples $u_i = u(x_i)$ for $i = 1, 2, \dots, N$ and is given by,

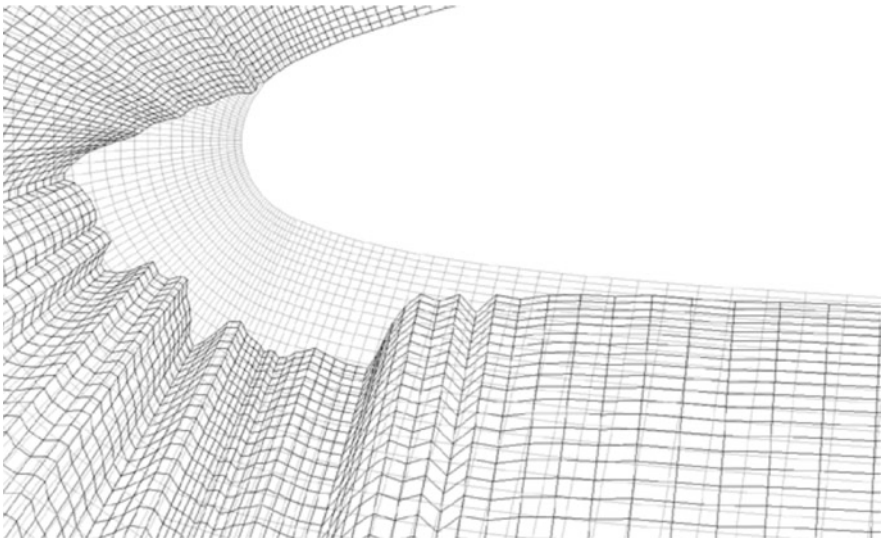


Fig. 15.12 Inverse Distance Weighting interpolation algorithm used to acquire the newly formed iced geometry in PoliMIce [60]. Here, the airfoil surface remains unmeshed while the original undeformed mesh follows it closely represented by the lighter lines. Once the surface is iced, however, the mesh deforms to follow the new profile and is represented by the darker lines

$$u(x) = \begin{cases} \frac{\sum_{i=1}^N w_i(x) U_i}{\sum_{i=1}^N w_i(x)}, & \text{if } d(x, x_i) \neq 0 \\ u_i, & \text{if } d(x, x_i) = 0, \end{cases} \quad (15.41)$$

where $w_i(x) = \frac{1}{d(x, x_i)^p}$ is a weighting function as defined by Shepard [89] and $d(x, x_i)$ represents the distance between the points x and x_i .

15.3.6 Numerical Results

The numerical results of the different fidelity levels of ice accretion simulations will now be compared against experimental data. These will also be displayed throughout the different ice regimes where different ice shapes are present and for large ice accretion time. Gori et al. [60] have demonstrated this with results from the PoliMice solver resembling closely to experimental data.

The first simulation was on a symmetric NACA0012 airfoil at zero angle of attack. The test conditions here were to assess icing during the rime ice regime and representative of winter conditions at low altitude. The results from this are shown in Fig. 15.13 and compared against that from the LEWICE code, the Myers model, and experimental data from Ruff et al. [90].

The second simulation was conducted with an angle of attack of 4° during the glaze ice regime and is shown in Fig. 15.14. This was also on a symmetric NACA0012 airfoil, however, the angle of attack now meant that the ice formation was not symmetric. It used the same experimental data [90].

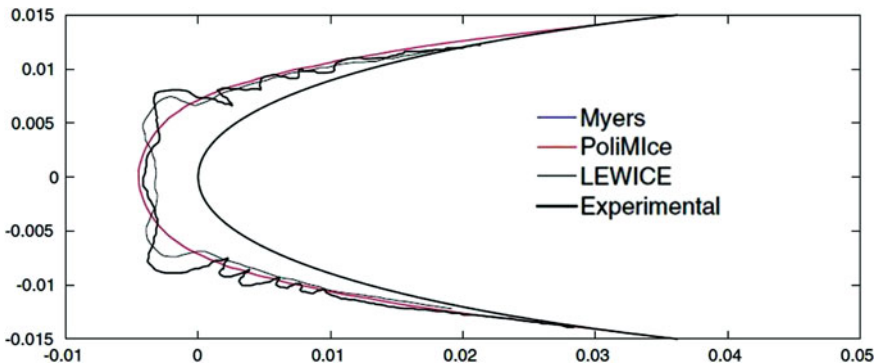


Fig. 15.13 Comparison of both PoliMice and LEWICE solvers and Myers model against experimental ice accretion shapes during the rime ice regime on a symmetric NACA0012 [60]

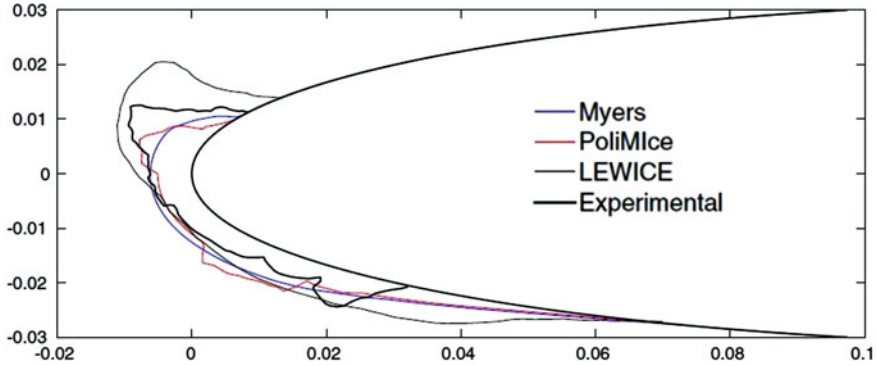


Fig. 15.14 Comparison of both PoliMIce and LEWICE solvers and Myers model against experimental ice accretion shapes during the glaze ice regime on a symmetric NACA0012 with an angle of attack of 4° [60]

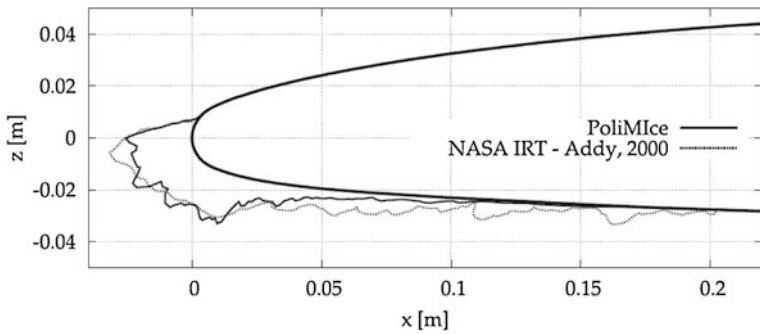


Fig. 15.15 Comparison of the PoliMIce solver and NASA IRT experimental ice accretion shapes at a long icing time on a GLC-305 airfoil at an angle of attack of 6° [60]

The third simulation was conducted over a long period of time meaning the ice thickness is large and is shown in Fig. 15.15. This simulation was compared against the experimental results presented by Addy et al. [91] in the NASA IRT wind tunnel facility. This simulation was conducted on a non-symmetric GLC-305 airfoil at an angle of attack of 6° .

15.4 Ice Protection Systems and Certification

The accretion of ice over aircraft surfaces has been found as a major hazard for the performance and safety of aircraft as it was presented in the Sect. 15.2. The UK’s Civil Aviation Authority [10] encouraged the pilots to have a clear situational awareness to avoid or minimize the exposure to icing conditions. However, in some occasions the exposure is unavoidable and consequently it is crucial to equip aircraft

with ice protection systems. Nowadays most of the aircraft include ice protection systems (IPS) to minimize the impact caused by the encounter of icing conditions in-flight.

Ice protection systems are deployed into aircraft to delay or to remove any ice accumulation on its surface and its components and therefore to maintain the performance and safety. The IPS can be classified into anti-icing or de-icing depending on whether they allow the ice to build on the aircraft. *Anti-icing* systems prevent the formation of ice, whereas *de-icing* technologies remove the already formed ice. They are implemented in several critical parts of the aircraft which are susceptible to build ice, such as the wings leading edge, the leading edge slats, the stabilizers, the engine nacelle, the wind screens, and the sensors [92]. Different technologies can be combined within the same aircraft for accomplishing the most convenient holistic protection. In this section, first the most mature IPS technologies, which are widely implemented [93–95], will be presented. Those can be classified into pneumatic thermal, de-icing boot, chemical, and electrical. Then, new IPS technologies will be briefly described and finally the regulations for certification to flight on icing conditions will be introduced.

15.4.1 Mature Protection Technologies

15.4.1.1 Pneumatic-Thermal Protection

This technology is widely used and it has been deployed in numerous commercial, defense, and business aircraft from different manufacturers such as Airbus A320 [93], Bombardier Challenger 300 [96], or Piaggio p.180 Avanti II [97]. Aircraft engines contain bleed air ports where pressurized hot air can be conveniently extracted from the compressor. A schematic view is presented in Fig. 15.16. The air

Fig. 15.16 Schematic view of a wing with a thermal anti-icing system. Source: FAA Aviation Maintenance Technician Handbook [92]

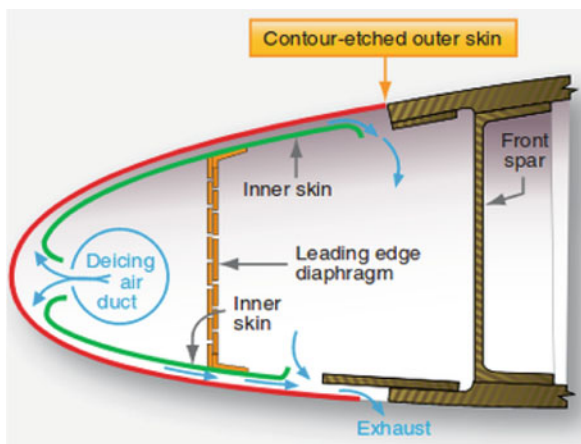
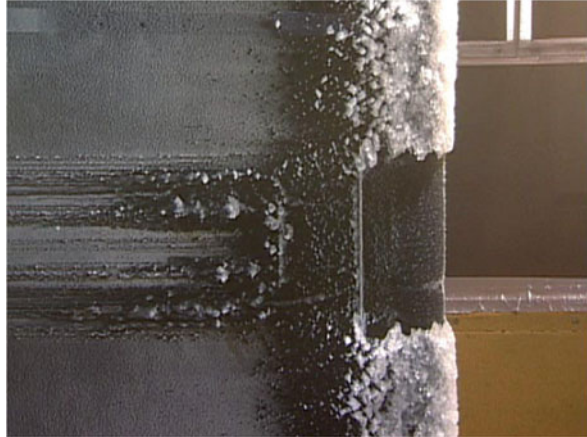


Fig. 15.17 Capture of runback ice occurring downstream the ice protection systems. Source: NASA



stream is rooted through pipes, manifolds, and valves to the required protected areas from leading edges or slats, stabilizers, and engine lips. The hot air is injected into a perforated pipe called piccolo tube (De-icing air duct in Fig. 15.16). The pressurized air moves across the perforations which are directed to the areas intended to be protected, heating the outer surface and melting the ice. In the case of the leading edge, it includes an inner skin layer through which the bleed air can escape after heating the surfaces. In addition, a temperature control is included together with an air intake to regulate the temperature within the leading edge. The availability and quality of bleed air depends on the power required by the engines and their size [98]. Thermal ice protection can be used as an anti-icing or de-icing technology depending on if it is activated before or after the ice is built.

Several research efforts focused on the mathematical modeling of those systems and their implementation of numerical models [99]. The protected areas are limited and when the formed ice is melted, the water runs downstream driven by the aerodynamic forces forming rivulets [100]. Since the temperature of the aircraft surfaces can be below freezing, the water might freeze aft in unprotected areas, and this can compromise the reliability and performance of the aircraft. The so-called runback water from anti-iced aircraft surfaces has been investigated [101, 102]. An example of a runback ice formation is depicted in Fig. 15.17. To mitigate the risk of freezing downstream some thermal protections systems are designed to fully or partially evaporate the impinging water. However, this technology requires a large energy consumption.

15.4.1.2 De-icing Boot Protection

This technology was first developed in 1930 [61], consequently it is a mature and proven technology. These systems are widely implemented in several commercial and business aircraft such as the Saab 2000 [95] and the TBM700 from Socata

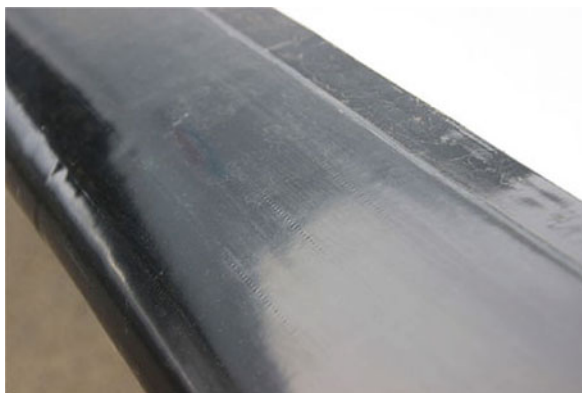
[103]. They are commonly deployed near the stagnation point on the leading edges, stabilizers, and propeller wings. The boots are normally oriented spanwise [104], although they can be deployed chord-wise or in certain critical surfaces. De-icing boot IPS consist of a set of sudden inflatable rubber boots positioned in the protected surfaces which are cyclically inflated and deflated according to the protection needs. They are de-icing systems because they remove an already formed layer of ice from the surfaces where they are implemented. Depending on the type of aircraft, boots are inflated with bleed air from the engines or an additional pneumatic system is included for that purpose [104]. Following the rupture of the ice, aerodynamic forces completely remove the ice from the surface [105]. The boots are made out of rubber and they require a supply of vacuum to avoid self-inflation. To improve the performance of these systems, chemical solutions that reduce the ice adhesion to the boot can be sprayed over the rubber boots.

Their activation time is critical. The more ice is accumulated in the surface, the more effective the system is. However, a compromise in performance and safety is to be found. In addition, late activation was claimed to cause the Comair flight 3272 fatal plane crash in 1997 [106]. An example of this technology is depicted in Fig. 15.18. Early activation of the boots it is a more conservative approach although it could cause “bridging.” Bridging occurs when ice dampens the sudden forces and remains attached. Nevertheless, this issue was solved by increasing the inflation pressure and letting a small layer of ice to accumulate. Residual ice can remain attached to the surfaces after the inflation and deflation of the boot which is eliminated in the subsequent cycles [61].

One of the drawbacks of the technology is the degradation of the rubber boots and consequently of the performance caused by interaction with extreme environmental conditions, such as very low temperatures, UV radiation, and atmospheric moisture. Therefore, very frequent maintenance and replacement is mandatory to ensure reliability and adequate performance. Current research is focused on materials with enhanced resistance to environmental factors.

Fig. 15.18 Pneumatic de-icing boot system attached to a wing leading edge.

Source: Wikipedia



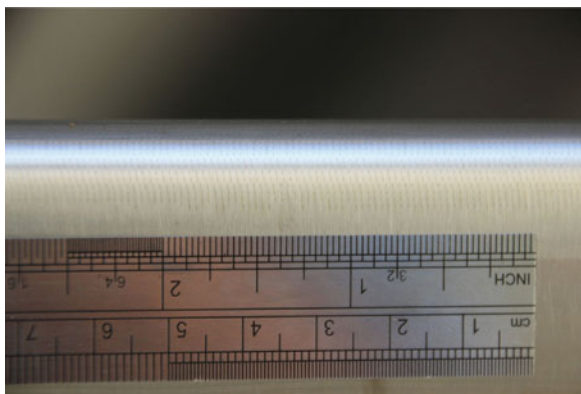
Research in the field of de-icing boot system was conducted to characterize the effects on aircraft performance of the residual and inter-cycle ice [107, 108]. For instance, Broeren conducted an experimental study based on wind tunnel tests using airfoils equipped with de-icing boots. The study concluded that the effects of the inter-cycle ice in the test conditions are more severe in terms of drag increase, lift decrease, and stall angle reduction than those of the residual ice after operating the de-icing boots. Plus, the authors observed that a little amount of ice was left on the airfoil after the inflation and deflation of the boots.

15.4.1.3 Thermo-Electric protection

These systems use the heat generated by an electric current going through a resistive component. The resistor component can present different schemes such as internal coil wires, externally wrapping blankets, tapes, conductive films, and heating rods. This technology has been implemented on the wing of the Boeing 787 [109]. They can be anti-icing systems if they do not allow the ice to build up or de-icing that operate when a layer of ice has been formed. De-icing systems break the adhesion between the surface of the aircraft and the built layer of ice. In a similar manner as the thermo-pneumatic systems, the electric ice protection systems can run fully evaporative or wet runback [110]. In the former approach, the impinging water is rapidly evaporated when impinging on the protected surfaces. In the latter approach, the ice is melted and the runback water is driven downstream by aerodynamic forces. The evaporative systems, however, require a larger power consumption. The resistive components can be mounted inside or outside the protected element or they can be built into it. This is the preferred method for the protection of rotorcraft [111] because it presents the best fit to the design, due to the relatively small thickness of the blades. In addition, electric protection means are implemented in the vast majority of aircraft to protect probes from icing such as the Pitot tube or the angle of attack probes [93, 94, 96, 97, 103]. Since the probes are usually small protuberances from the aircraft surface, they are very susceptible to ice.

Numerical and mathematical models [112, 113] have been developed in order to increase the understanding of the performance of the protection systems and to accurately predict the effects. The goal is to forecast possible effects and to help on the design of more effective ice protection systems. Moreover, some authors investigated the use of new conductive materials to improve the efficiency of the systems and the combination of thermal-electrical systems with super-hydrophobic coatings to decrease the energy consumption [28]. Also, layers of carbon nano-tubes have been investigated due to their electrical properties such as high current density and low heat capacity [114].

Fig. 15.19 Micro-perforated leading edge of an aircraft protected with chemical anti-icing system Source: Wikimedia commons. Source: Wikimedia commons



15.4.1.4 Chemical Protection

This technology was also deployed in the Hawker 800XP from Raytheon [94]. Chemical systems are usually implemented to protect leading edges of the wings, stabilizers, wind shields, and propellers. These solutions (mainly glycol based [115]) are flushed into the protected areas to reduce the fusion temperature of water. They can be used as anti-icing or de-icing systems depending on if the solution is distributed before or after the ice formation. When operating in de-icing mode, the solution breaks the adhesion of the ice from the aircraft surface which is then removed by aerodynamic forces. When operating in anti-icing mode, the solution prevents the liquid droplets to freeze so the water is advected downstream by aerodynamic forces. In the case of the leading edge, a micro-perforated panel is attached to/ built into the leading edge whose perforated diameters can measure $63\ \mu\text{m}$ [104]. Behind the panel there is a reservoir created by a back panel where the solution is stored within an external tank and it is pumped by means of an electrical pump. A membrane is attached to the porous surface to evenly distribute the solution over the surface of the porous material. When the system is activated, the solution is pumped to the reservoir and distributed outwards through the micro-perforations. In Fig. 15.19, one can see a micro-perforated leading edge of an aircraft wing. These systems are known as wiping wing or TKS in reference to the trade mark of the systems. One of the drawbacks is that the aircraft protection is dependent on the availability of the solution, that is when the solution is finished the aircraft is unprotected. In addition, the tank and the solution increase the weight of the aircraft.

15.4.2 Alternative Protection Technologies

The described mature ice protection systems present several drawbacks. There are alternative technologies available and under development which are meant to provide more reliable and cost-effective solutions. These technologies take

advantage of special properties of the materials or physical principles. Regarding the special properties, super-hydrophobic materials are investigated for anti-icing and de-icing purposes to decrease the adhesion of the impinging droplets on aerodynamic surfaces [116]. In addition, graphite is studied as well due to its high thermal inertia [117]. Furthermore, shape memory alloys change their shape when energy is supplied to them, generally external heating. This property has been exploited to crack a surface of ice in a similar manner as the de-icing boots [117]. Electromagnetic forces have been investigated to induce sudden motions on elements that hit the protected surface and detach the ice [118]. These forces are investigated to modify the shape of a surface in a similar manner as the shape memory form alloys [117]. Finally, ultrasounds are under investigation [119]. Sound waves are used to create stress on the layer of ice and to detach it from the surface.

15.4.3 Regulations and Certification

The regulations related to ice accretion seek to ensure the safe and reliable operation of aircraft equipped with ice protection systems when those are exposed to atmospheric icing conditions. On another note, an ice envelope consists of a combination of atmospheric parameters that describe an atmospheric icing scenario such as LWC, MVD, and OAT and cloud length. Certification for aircraft effective protection against icing effects is attained by ensuring the aircraft can fly safely through a predetermined set of icing envelopes. Next the most relevant regulations and certifications are presented.

15.4.3.1 FAA Code of Flight Regulations

The American Federal Aviation Administration (FAA) belongs to the U.S. department of transportation whose mission is to ensure fast, safe, reliable, and efficient transport. Within the FAA, the Aviation Safety department (AVS) is in charge of elaborating regulations for the certification of commercial and civil aircraft, namely Code of Federal Regulation (CFR). The regulations for flight in icing conditions can be found in the section 25 of the chapter 14 of the CFR. The Appendix C of this section presents a range of icing envelopes that aircraft should be protected to, which are divided into two:

- **Continuous maximum icing:** Set of icing envelopes which characterize long duration encounters found when aircraft fly through stratiform clouds. These clouds, as presented in the Sect. 15.2.1.1, generally present low liquid water content and largely extend in the horizontal direction. The values of the atmospheric parameters can be found from the graphs depicted in Figs. 15.20 and 15.21, for a reference cloud length of 17.4 nautical miles. In addition, on the regulation one

Fig. 15.20 Atmospheric icing conditions for continuous maximum exposure. Outside air temperature (OAT) and pressure altitude. Source: FAA Regulations

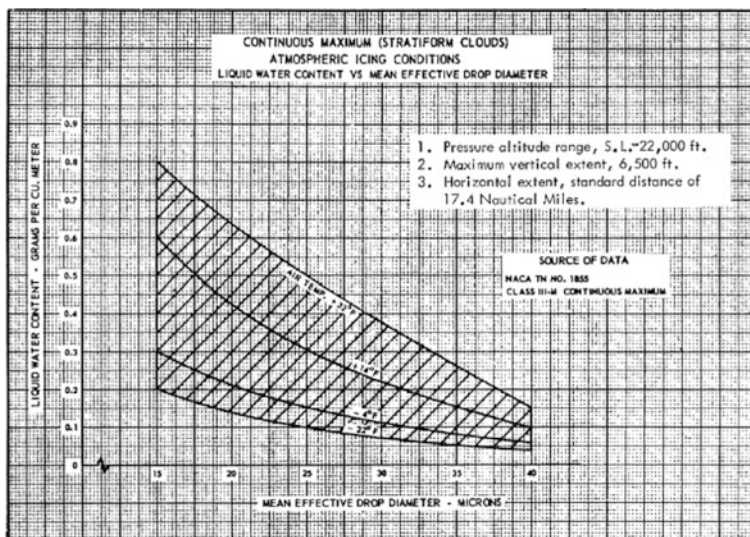
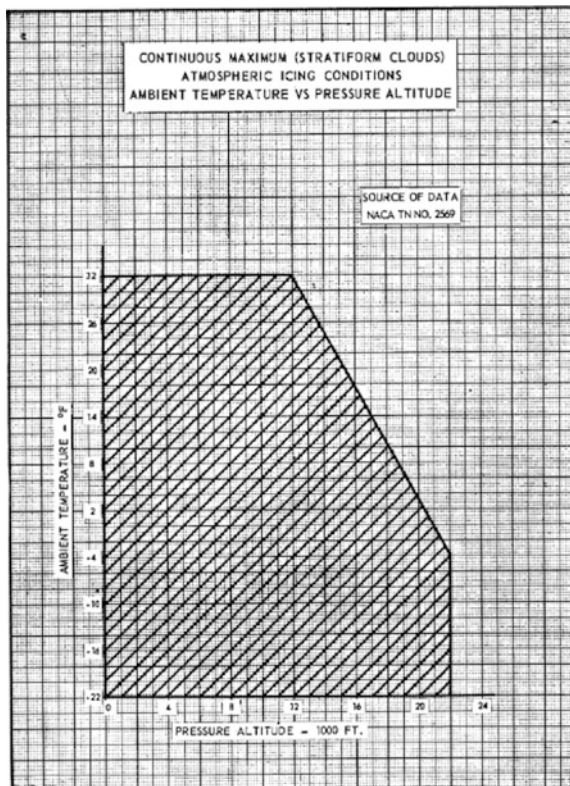


Fig. 15.21 Atmospheric icing envelopes for continuous maximum exposure. Mean Volume Diameter (MVD), Liquid water content (LWC), and Outside air temperature (OAT). Source: FAA Regulations

can find an additional a procedure to extrapolate the icing envelopes to any cloud length.

- Intermittent maximum icing: Icing envelopes that characterize severe short encounters occurring when the aircraft flies through cumulus clouds. Cumulus clouds extend vertically and commonly contain high liquid water content. Large amounts of ice accreted on the surface of the aircraft are expected. The values of the atmospheric parameters can be found in the regulations code which are analogous to Figs. 15.20 and 15.21.

In addition, Appendix C presents aircraft operation stages that must be studied from the icing effects point of view. To reduce the number of tests required while ensuring the protection under all the envelopes, it is accepted the study of the most critical conditions. The Appendix O includes SLD icing envelopes for droplets larger than $40\ \mu\text{m}$ in a similar manner as it was presented in Figs. 15.20 and 15.21. The icing envelopes include freezing drizzle and freezing rain conditions, which are found when an aircraft is flying through or below a stratiform cloud. The fulfilment of the protection against SLD is critical since the SLD accretions have been claimed the cause of aircraft crashes.

To attain certification to fly in icing conditions, an aircraft must be able to “safely operate” in both maximum continuous and intermittent conditions and in freezing rain and drizzle. The full description of the conditions is described in Sections 25.1419 and 25.1420. The “safe operation” should be ensured in all the phases of flight, including take off, holding, descent, landing and go-around. This must be assessed by in-flight or laboratory test of aircraft, parts or models exposed to the critical ice envelopes presented in the Appendixes C and O. In these tests, the ice protection systems must be operated according to the operations flight manual and ice detections systems or procedures must be available. Additionally, in the case of SLD ice encounters, the aircraft must be able to safely exit them.

Snow and mixed icing conditions are included in a set of envelopes presented in the appendix D to Part 33. Those are defined by the altitude of the encounter, the total water content (TWC) in g/m^3 , and the ice crystal size median mass dimension (MMD). Section 25.1093 states that the aircraft must be able to operate in all the envelopes presented in Appendixes C, D, and O and in the presence of falling or blowing snow. In addition, the aircraft design must ensure the avoidance of ice accumulations that lead to faulty engine operation, loss of power or stall.

15.4.3.2 EASA Certification Specifications

The European Aviation Safety Agency (EASA) is accountant for all the European civilian aviation safety. Among their missions is the aircraft certification for all the state members and collaboration with other international agencies to promote worldwide standards and achieve the highest safety level. The corresponding certification department developed a set of regulations to ensure aircraft reliability and safety when encountering icing conditions, which is described in section 25.

The European regulations are found to be analogous to the FAA ones presented in the Sect. 15.4.3.1. The icing envelopes are presented in appendices C, D, and O and the regulation and certification it is described in the sections 25.1419, 25.1420, and 25.1093.

15.5 Concluding Remarks

An overview of the in-flight icing phenomena has been presented, including its physics, research methods, and ice protection systems. Within the research methods the replication of ice shapes, whether experimental, numerical, or analytical is subject to uncertainties. Input data for models gathered in-flight presents uncertainties related to the accuracy of the measuring devices. For instance, the error of the King probe, which measures the cloud mean volume diameter, can reach up to 30%. Although the technology of the measuring devices has evolved, some published data was gathered in field campaigns several years ago. This data still needs to be used since, in some cases, it is the only data available. It has been used for validation purposes to replicate icing shapes, as model inputs or to elaborate regulation to derive possible existing icing envelopes. The uncertainties need to be addressed, to measure their individual and overall effect on the discrepancies between models and real phenomena.

Physical reproduction of in-flight icing phenomena in icing tunnel tests presents also uncertainties related to the replication of actual airflows and cloud properties. The cloud-related uncertainties are also applicable to in-flight tests in which the studied aircraft is preceded by a water tank that sprays water droplets. It was found in literature that the replicability of the same icing test could reach up to 30% in some cases, when ideally it should be 0%. Some of these uncertainties can be reduced by further testing, regular calibration of the devices and retrofitting facilities and devices, however, this implies further efforts.

There has been a huge progress on the numerical modeling of in-flight icing as it has been presented in the part Sect. 15.3. Some of the numerical models present really good agreement with experimental results. However, additional studies are required to capture intricate ice shapes. Further work would entail the quantification of the uncertainties in model inputs and in analytical and numerical models themselves to assess the global uncertainty of the results. Then, the uncertainties could be rated depending on their importance in the overall uncertainty of the results and further efforts could be conducted in the reduction of the largest epistemic uncertainties. In addition, input and model uncertainties could be propagated through the models to predict the uncertainty of the modeling results and elaborate more robust predictions. Improved capture of the in-flight icing phenomena can lead to the design of more robust ice protection systems, which can ensure safety still in a more efficient manner.

References

1. M. Bragg, Aircraft aerodynamic effects due to large droplet ice accretions, in *34th Aerospace Sciences Meeting and Exhibit* (1996), p. 932
2. J. Marwitz, M. Politovich, B. Bernstein, F. Ralph, P. Neiman, R. Ashenden, J. Bresch, Meteorological conditions associated with the ATR72 aircraft accident near Roselawn, Indiana, on 31 October 1994. *Bull. Am. Meteorol. Soc.* **78**(1), 41–52 (1997)
3. P. Galison, An accident of history, in *Atmospheric Flight in the Twentieth Century* (Springer, Berlin, 2000), pp. 3–43
4. T.A.S. Council, In-flight icing encounter and crash into the sea TransAsia airways flight 791ATR72-200, B-2270817 kilometers southwest of Makung city, Penghu Islands. Occurrence Investigation (Report No. ASC-AOR-05-04-001). Aviation Safety Council, Taipei (2002)
5. Y. Cao, Z. Wu, Y. Su, Z. Xu, Aircraft flight characteristics in icing conditions. *Prog. Aerosp. Sci.* **74**, 62–80 (2015)
6. F.T. Lynch, A. Khodadoust, Effects of ice accretions on aircraft aerodynamics. *Progr. Aerosp. Sci.* **37**(8), 669–767 (2001)
7. A. Cavagna, Supercooled liquids for pedestrians. *Phys. Rep.* **476**(4–6), 51–124 (2009)
8. K. Saha, *The Earth's Atmosphere. Its Physics and Dynamics* (Springer Science & Business Media, Berlin, 2008)
9. B. Stevens, *Twelve Lectures on Cloud Physics* (Max Planck Institute for Meteorology-University of Hamburg, Hamburg, 2010)
10. C.A. Authority, *Aircraft Icing Handbook* (Safety Education and Publishing Unit, New Zealand, 2000), p. 2
11. J. Ballough, Pilot's guide: Flight in icing conditions, FAA, AC, no. 91–74A (2007)
12. R. Gent, N. Dart, J. Cansdale, Aircraft icing. *Philos. Trans. R. Soc. Lon. A Math. Phys. Eng. Sci.* **358**(1776), 2873–2911 (2000)
13. M. Politovich, Aircraft icing. *Encycl. Atmos. Sci.* **358**(1776), 68–75 (2003)
14. T. Cebeci, F. Kafyeke, Aircraft icing. *Ann. Rev. Fluid Mech.* **35**, 11–21 (2003)
15. J. Mason, W. Strapp, P. Chow, The ice particle threat to engines in flight, in *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006), p. 206
16. R.P. Lawson, L.J. Angus, A.J. Heymsfield, Cloud particle measurements in thunderstorm anvils and possible weather threat to aviation. *J. Aircr.* **35**(1), 113–121 (1998)
17. J.J.P. Veres, P.C. Jorgenson, Modeling commercial turbofan engine icing risk with ice crystal ingestion, in *5th AIAA Atmospheric and Space Environments Conference* (2013), p. 2679
18. L. Makkonen, Models for the growth of rime, glaze, icicles and wet snow on structures. *Philos. Trans. R. Soc. Lon. A Math. Phys. Eng. Sci.* **358**(1776), 2913–2939 (2000)
19. B.E. Kringlebotn Nygaard, H. Ágústsson, K. Somfalvi-Tóth, Modeling wet snow accretion on power lines: improvements to previous methods using 50 years of observations. *J. Appl. Meteorol. Climatol.* **52**(10), 2189–2203 (2013)
20. Y. Sakamoto, Snow accretion on overhead wires. *Philos. Trans. Royal Soc. London A Math. Phys. Eng. Sci.* **358**(1776), 2941–2970 (2000)
21. W. King, D. Parkin, R. Handsworth, A hot-wire liquid water device having fully calculable response characteristics. *J. Appl. Meteorol.* **17**(12), 1809–1813 (1978)
22. P. Tran, M. Brahim, I. Paraschivoiu, A. Pueyo, F. Tezok, Ice accretion on aircraft wings with thermodynamic effects. *J. Aircr.* **32**(2), 444–445 (1995)
23. H. Beaugendre, A PDE-Based 3D approach to in-flight ice accretion. Ph.D. Thesis, Mechanical Engineering Department, McGill University, Montreal (2003)
24. K.J. Finstad, E.P. Lozowski, L. Makkonen, On the median volume diameter approximation for droplet collision efficiency. *J. Atmos. Sci.* **45**(24), 4008–4012 (1988)
25. D. Baumgardner, An analysis and comparison of five water droplet measuring instruments. *J. Climate Appl. Meteorol.* **22**(5), 891–910 (1983)

26. C. Antonini, A. Amirfazli, M. Marengo, Drop impact and wettability: from hydrophilic to superhydrophobic surfaces. *Phys. Fluids* **24**(10), 102104 (2012)
27. D. Mangini, C. Antonini, M. Marengo, A. Amirfazli, Runback ice formation mechanism on hydrophilic and superhydrophobic surfaces. *Cold Regions Sci. Technol.* **109**, 53–60 (2015)
28. C. Antonini, M. Innocenti, T. Horn, M. Marengo, A. Amirfazli, Understanding the effect of superhydrophobic coatings on energy reduction in anti-icing systems. *Cold Regions Sci. Technol.* **67**(1–2), 58–67 (2011)
29. L. Raraty, D. Tabor, The adhesion and strength properties of ice. *Proc. R. Soc. Lond. A* **245**, 184–201 (1958)
30. L. Cao, A.K. Jones, V.K. Sikka, J. Wu, D. Gao, Anti-icing superhydrophobic coatings. *Langmuir* **25**(21), 12444–12448 (2009)
31. A.J. Meuler, J.D. Smith, K.K. Varanasi, J.M. Mabry, G.H. McKinley, R.E. Cohen, Relationships between water wettability and ice adhesion. *ACS Appl. Mater. Interfaces* **2**(11), 3100–3110 (2010)
32. R. Scavuzzo, M. Chu, V. Ananthaswamy, Influence of aerodynamic forces in ice shedding. *J. Aircr.* **31**(3), 526–530 (1994)
33. S. Zhang, O. El Kerdi, R.A. Khurram, W.G. Habashi, Fem analysis of in-flight ice break-up. *Finite Elem. Anal. Des.* **57**, 55–66 (2012)
34. M. Papadakis, H.W. Yeong, I.G. Soares, J. Jacob, Experimental and computational investigation of ice shedding from aircraft surfaces, in *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006)
35. A.L. Yarin, Drop impact dynamics: splashing, spreading, receding, bouncing. . . . *Annu. Rev. Fluid Mech.* **38**, 159–192 (2006)
36. R. Rioboo, C. Tropea, M. Marengo, Outcomes from a drop impact on solid surfaces. *At. Sprays* **11**(2), 12 (2001)
37. A. Yarin, D. Weiss, Impact of drops on solid surfaces: self-similar capillary waves, and splashing as a new type of kinematic discontinuity. *J. Fluid Mech.* **283**, 141–173 (1995)
38. P. Brambilla, A. Guardone, Assessment of dynamic adaptive grids in volume-of-fluid simulations of oblique drop impacts onto liquid films, *J. Comput. Appl. Math.* **281**, 277–283 (2015)
39. P. Gondret, E. Hallouin, M. Lance, L. Petit, Experiments on the motion of a solid sphere toward a wall: from viscous dissipation to elasto-hydrodynamic bouncing. *Phys. Fluids* **11**(9), 2803–2805 (1999)
40. A.A. Katak, R.H. Davis, Oblique collisions and rebound of spheres from a wetted surface. *J. Fluid Mech.* **509**, 63–81 (2004)
41. R. Kind, M. Potapczuk, A. Feo, C. Golia, A. Shah, Experimental and computational simulation of in-flight icing phenomena. *Prog. Aerosp. Sci.* **34**(5–6), 257–345 (1998)
42. M.B. Bragg, A.P. Broeren, L.A. Blumenthal, iced-airfoil aerodynamics. *Prog. Aerosp. Sci.* **41**(5), 323–362 (2005)
43. S. Lee, M.B. Bragg, Experimental investigation of simulated large-droplet ice shapes on airfoil aerodynamics. *J. Aircr.* **36**(5), 844–850 (1999)
44. H. Gurbachi, M. Bragg, Sensing aircraft icing effects by flap hinge moment measurement, in *17th Applied Aerodynamics Conference* (2000), p. 3149
45. M. Vargas, M. Papadakis, M. Potapczuk, H. Addy, D. Sheldon, J. Giriunas, Ice Accretions on a Swept GLC-305 Airfoil. *SAE Transactions*, 58–71 (2002)
46. E.A. Whalen, A.P. Broeren, M.B. Bragg, Aerodynamics of scaled runback ice accretions. *J. Aircr.* **45**(2), 591–603 (2008)
47. E. Arrington, M. Pickett, D. Sheldon, Flow quality studies of the NASA Lewis Research Center Icing Research Tunnel, in *25th Plasmadynamics and Lasers Conference* (1994), p. 2590
48. M. Zocca, G. Gori, A. Guardone, Blockage and three-dimensional effects in wind-tunnel testing of ice accretion over wings. *J. Aircr.* **54**(2), 759–767 (2016)

49. M. Papadakis, G. Zumwalt, R. Elangonan, G. Freund Jr, M. Breer, L. Whitmer, An experimental method for measuring water droplet impingement efficiency on two-and three-dimensional bodies, in *24th Aerospace Sciences Meeting* (1989)
50. M. Papadakis, K.E. Hung, G.T. Vu, H.W. Yeong, C.S. Bidwell, M.D. Breer, T.J. Bencic, *Experimental Investigation of Water Droplet Impingement on Airfoils, Finite Wings, and an S-duct Engine Inlet* (National Aeronautics and Space Administration, Glenn Research Center, Cleveland Ohio, 2002)
51. Y. Cao, C. Ma, Q. Zhang, J. Sheridan, Numerical simulation of ice accretions on an aircraft wing. *Aerosp. Sci. Technol.* **23**(1), 296–304 (2012)
52. J. Tsao, A. Rothmayer, A mechanism for ice roughness formation on an airfoil leading edge, contributing to glaze ice accretion, in *36th AIAA Aerospace Sciences Meeting and Exhibit* (1998), p. 485
53. G. Fortin, A. Ilinca, J.-L. Laforte, V. Brandi, New roughness computation method and geometric accretion model for airfoil icing. *J. Aircr.* **41**(1), 119–127 (2004)
54. R.Z. Blackmore, E.P. Lozowski, A theoretical spongy spray icing model with surficial structure. *Atmos. Res.* **49**(4), 267–288 (1998)
55. R.Z. Blackmore, L. Makkonen, E. Lozowski, A new model of spongy icing from first principles. *J. Geophys. Res. Atmos.* **107**(D21), AAC-9 (2002)
56. W. Kong, H. Liu, Development and theoretical analysis of an aircraft supercooled icing model. *J. Aircr.* **51**, 975–986 (2014)
57. I. Nomenclature, Further refinement of the LEWICE SLD model, in *44th AIAA Aerospace Sciences Meeting and Exhibit* (2006)
58. T. Hedde, D. Guffond, Onera three-dimensional icing model. *AIAA J.* **33**(6), 1038–1045 (1995)
59. S. Nilamdeen, W. Habashi, M. Aube, G. Baruzzi, FENSAP-ICE: Modeling of water droplets and ice crystals, in *Proceedings of the 1st AIAA Atmospheric and Space Environments Conference* (2009)
60. G. Gori, M. Zocca, M. Garabelli, A. Guardone, G. Quaranta, PoliMice: a simulation framework for three-dimensional ice accretion. *Appl. Math. Comput.* **267**, 96–107 (2015)
61. G. Mingione, M. Barocco, Flight in icing conditions summary, French DGAC (1997)
62. S.G. Pouryoussefi, M. Mirzaei, M.-M. Nazemi, M. Fouladi, A. Doostmahmoudi, Experimental study of ice accretion effects on aerodynamic performance of an NACA 23012 airfoil. *Chinese J. Aeronaut.* **29**(3), 585–595 (2016)
63. M.B. Bragg, Effects of ice accretion on aircraft aerodynamics (1998)
64. K. Rosen, M. Potash, Forty years of helicopter ice protection experience at Sikorsky aircraft. *J. American Helicopter Soc.* **26**(3), 5–19 (1981)
65. K. Korkan, E. Cross Jr, T. Miller, Performance degradation of a model helicopter rotor with a generic ice shape. *J. Aircr.* **21**(10), 823–830 (1984)
66. K.D. Korkan, L. Dadone, R.J. Shaw, Performance degradation of helicopter rotor in forward flight due to ice. *J. Aircr.* **22**(8), 713–718 (1985)
67. R.K. Jeck, Icing design envelopes (14 CFR parts 25 and 29, Appendix c) converted to a distance-based format, Technical Report, Federal Aviation Administration Technical Center Atlantic City New Jersey (2002)
68. F. Palacios, J. Alonso, K. Duraisamy, M. Colonno, J. Hicken, A. Aranake, A. Campos, S. Copeland, T. Economon, A. Lonkar, et al., Stanford university unstructured (SU 2): An open-source integrated computational environment for multi-physics simulation and design, in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (2013), p. 287
69. H. Jasak, A. Jemcov, Z. Tukovic, et al., OpenFOAM: A C++ library for complex physics simulations, in *International Workshop on Coupled Methods in Numerical Dynamics*, vol. 1000 (IUC Dubrovnik, Croatia, 2007), pp. 1–20
70. W.B. Wright, Users manual for the improved NASA Lewis ice accretion code LEWICE 1.6 (1995)
71. T. Cebeci, Calculation of flow over iced airfoils. *AIAA J.* **27**(7), 853–861 (1989)

72. J. Chung, A. Reehorst, Y. Choo, M. Potapczuk, J. Slater, Navier-Stokes analysis of flowfield characteristics of an ice-contaminated aircraft wing. *J. Aircr.* **37**(6), 947–959 (2000)
73. J.C. Warner, C.R. Sherwood, H.G. Arango, R.P. Signell, Performance of four turbulence closure models implemented using a generic length scale method. *Ocean Model.* **8**(1–2), 81–113 (2005)
74. F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA J.* **32**(8), 1598–1605 (1994)
75. T.A. Dunn, E. Loth, M.B. Bragg, Computational investigation of simulated large-droplet ice shapes on airfoil aerodynamics. *J. Aircraft* **36**(5), 836–843 (1999)
76. S. Lee, T. Dunn, H. Gurbachi, M. Bragg, E. Loth, An experimental and computational investigation of spanwise-step-ice shapes on airfoil aerodynamics, in *36th AIAA Aerospace Sciences Meeting and Exhibit* (1998), p. 490
77. S. Marques, K. Badcock, J. Gooden, S. Gates, W. Maybury, Validation study for prediction of iced aerofoil aerodynamics. *Aeronaut. J.* **114**(1152), 103–111 (2010)
78. G. Parma, A model for in-flight ice accretion based on the exact solution of the unsteady stefan problem, Master’s Thesis, Politecnico di Milano, 2015
79. M. Tribus, G. Young, L. Boelter, Analysis of heat transfer over a small cylinder in icing conditions on Mount Washington. *Trans. ASME* **70**(8), 971–976 (1948)
80. M. Bragg, S. Lee, C. Henze, M. Bragg, S. Lee, C. Henze, Heat-transfer and freestream turbulence measurements for improvement of the ice accretion physical model, in *35th Aerospace Sciences Meeting and Exhibit* (1997), p. 53
81. B.L. Messinger, Equilibrium temperature of an unheated icing surface as a function of air speed. *J. Aeronaut. Sci.* **20**(1), 29–42 (1953)
82. I. Langmuir, K. Blodgett, et al., *Mathematical Investigation of Water Droplet Trajectories* (Distributed by Office of the Publication Board, Washington, 1946)
83. Y. Bourgault, W. Habashi, J. Dompierre, Z. Boutanos, W. Di Bartolomeo, An Eulerian approach to supercooled droplets impingement calculations, in *AIAA Paper-97-0176* (1997)
84. R. Gent, Calculation of water droplet trajectories about an aerofoil in steady, two-dimensional, compressible flow, RAE Technical Report, vol. 84060 (1984)
85. G.K. Batchelor, *An introduction to Fluid Dynamics* (Cambridge University Press, Cambridge, 2000)
86. T.G. Myers, Extension to the messenger model for aircraft icing. *AIAA J.* **39**(2), 211–218 (2001)
87. C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.* **135**(2), 203–216 (1997)
88. I. Sazonov, D. Wang, O. Hassan, K. Morgan, N. Weatherill, A stitching method for the generation of unstructured meshes for use with co-volume solution techniques. *Comput. Methods Appl. Mech. Eng.* **195**(13), 1826–1845 (2006)
89. D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in *Proceedings of the 1968 23rd ACM National Conference* (ACM, New York, 1968), pp. 517–524
90. G.A. Ruff, B.M. Berkowitz, Users manual for the NASA Lewis ice accretion prediction code (LEWICE) (1990)
91. H.E. Addy Jr, Ice accretions and icing effects for modern airfoils, Technical Report, National Aeronautics and Space Administration Cleveland OH Glenn Research Center (2000)
92. *Aviation Maintenance Technician Handbook- Airframe*, vol. 2, ch. Chapter 15. U.S. Department of Transportation. Federal Aviation Administration (2012), pp. 15.1–15.32
93. Airbus, *Ice and Rain Protection. Flight Crew Operational Manual Airbus A320 Simulator* (1987)
94. Raytheon Aircraft, *Hawker 800XP Pilot’s Operating Manual. Systems Description* (2002)
95. Saab, *Saab 2000 Aircraft Operations Manual* (1998)
96. Bombardier, *Bombardier Challenger 300 Flight Crew Operating Manual*, Ice and Rain Protection ed. (2005)
97. Piaggio, *P 180 Avanti II Pilot’s Operating Handbook* (2006)

98. K. Yeoman, Efficiency of a bleed air powered inlet icing protective system, in *AIAA Paper*, vol. 717 (1994), p. 1994
99. F. Morency, F. Tezok, and I. Paraschivoiu, Anti-icing system simulation using CANICE. *J. Aircr.* **36**(6), 999–1006 (1999)
100. G. Silva, O. Silvaes, E. Zerbini, Water film breakdown and rivulets formation effects on thermal anti-ice operation simulation, in *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference* (2006), p. 3785
101. K.M. Al-Khalil, T.G. Keith Jr, K.J. De Witt, New concept in runback water modeling for anti-iced aircraft surfaces. *J. Aircr.* **30**(1), 41–49 (1993)
102. G. Silva, O. Silvaes, E. Zerbini, Airfoil anti-ice system modeling and simulation, in *41st Aerospace Sciences Meeting and Exhibit* (2003), p. 734
103. Socata, *TBM700 Pilot's Operating Handbook. Ice Protection System* (2010)
104. Aircraft icing handbook, Technical Report vol. 2, FAA Technical Center (1991)
105. Give ice the boot, Technical Report, UTC Aerospace Systems (2016)
106. N. T. Safety Board, Aircraft accident report in-flight encounter and uncontrolled collision with terrain. Comair flight 3272 Embraer Monroe, Michigan, in *Safety*, vol. 800 (National Transportation Safety Board, Washington 1997), pp. 877–6799
107. A.P. Broeren, M.B. Bragg, H.E. Addy, Effect of intercycle ice accretions on airfoil performance. *J. Aircr.* **41**(1), 165–174 (2004)
108. A.P. Broeren, M.B. Bragg, Effect of airfoil geometry on performance with simulated intercycle ice accretions. *J. Aircr.* **42**(1), 121–130 (2005)
109. M. Sinnott, 787 no-bleed systems: saving fuel and enhancing operational efficiencies, in *A Quarterly Publication Boeing Aeromagazine* (2007)
110. R. Foster, Aircraft electrical ice protection systems. *Aircraft Eng. Aerosp. Technol.* **40**(7), 7–8 (1968)
111. H. Coffman, Helicopter rotor icing protection methods. *J. Am. Helicopter Soc.* **32**(2), 34–39 (1987)
112. G.D. Silva, O.D.M. Silvaes, E.D.J. Zerbini, Numerical simulation of airfoil thermal anti-ice operation part 1: mathematical modeling. *J. Aircr.* **44**(2), 627–634 (2007)
113. G.d. Silva, O.d.M. Silvaes, E.d.G. Zerbini, Numerical simulation of airfoil thermal anti-ice operation part 2: implementation and results. *J. Aircr.* **44**(2), 635 (2007)
114. S.T. Buschhorn, S.S. Kessler, N. Lachmann, J. Gavin, G. Thomas, B.L. Wardle, Electrothermal icing protection of aerosurfaces using conductive polymer nanocomposites, in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* (2013), p. 1729
115. D. Kohlman, W. Schweikhard, P. Evanich, Icing-tunnel tests of a glycol-exuding, porous leading-edge ice protection system. *J. Aircraft* **19**, 647–654 (1982)
116. T. Wang, Y. Zheng, A.-R.O. Raji, Y. Li, W.K. Sikkema, J.M. Tour, Passive anti-icing and active deicing films. *ACS Appl. Mater. Interfaces* **8**(22), 14169–14173 (2016)
117. Z. Goraj, An overview of the deicing and anti-icing technologies with prospects for the future, in *24th International Congress of the Aeronautical Sciences*, vol. 29 (2004)
118. M. Endresi, H. Sommerwerkd, C. Mendigu, M. Sinapiusi, P. Horstd, Experimental study of two mechanical de-icing systems applied on a wing section tested in an icing wind tunnel. *CEAS Aeronaut. J.* **8**, 429–439 (2016)
119. M. Budinger, V. Pommier-Budinger, G. Napias, A.C. Da Silva, Ultrasonic ice protection systems: analytical and numerical models for architecture tradeoff. *J. Aircr.* **53**(3), 680–690 (2016)

Chapter 16

Uncertainty Treatment Applications: High-Enthalpy Flow Ground Testing



Anabel del Val, Olivier Chazot, and Thierry Magin

Mach number is like aborigine counting: one, two, three, four, many. Once you reach many, the flow is hypersonic.

H. K. Beckmann

Abstract In this chapter, the operating principles of two major types of high-enthalpy facilities are reviewed. Both groups of facilities are complementary to reproduce high-speed flows and high thermal loads. The description focuses on the main elements and basic functioning of these wind tunnels, as well as the measurement techniques involved, including an analysis of experimental uncertainties. The physico-chemical models are then introduced and explained in detail giving an account of the complexities facing uncertainty quantification methods applied to this particular system. Computational tools to simulate flow conditions in inductively-coupled plasma facilities are described in detail providing mesh examples of the different fields involved. Finally, conclusions and remarks concerning uncertainty quantification challenges for atmospheric entry flows are discussed with special emphasis on the multi-physics and high-dimensionality aspects of the system. A recount on the state of the art on safety margins is given as well.

Keywords High-enthalpy flows · Uncertainty quantification · Atmospheric entry · Ground testing

16.1 Atmospheric Entry: A Complex Problem

After the successful crewed missions to the Moon and many probe entries into the atmosphere of outer planets, the next challenges of space exploration include bringing samples back to Earth by means of robotic missions, as well as continuing

A. del Val (✉) · O. Chazot · T. Magin
von Karman Institute for Fluid Dynamics, Rhode-St-Genèse, Belgium
e-mail: delvalbeni@vki.ac.be; chazot@vki.ac.be; magin@vki.ac.be

© Springer Nature Switzerland AG 2021
M. Vasile (ed.), *Optimization Under Uncertainty with Applications to Aerospace Engineering*, https://doi.org/10.1007/978-3-030-60166-9_16

507

the crewed space program to send human beings to Mars and bring them home safely. Space agencies are also facing technological problems when dealing with residues of space flight. More than 30,000 space debris constitute today threats for space systems in orbit, such as the International Space Station and active satellites for observation and communication, as well as for humankind when debris, not fully destroyed during re-entry, impact the Earth. Facing these problems requires in-depth knowledge of the systems and phenomena involved.

Orbital velocities were predicted by Newton in the seventeenth century when conceiving the equations for motion under gravitational forces. These equations imply the existence of a velocity threshold above which an object could *fly* forever (ideally). Low Earth Orbits (LEO) are the most common among satellites and space stations with an average orbital velocity of 7 km/s. Traveling beyond the orbit of our planet means more energy, interplanetary velocities are of the order of 11 km/s. All this amount of kinetic and potential energy, dictated by orbital mechanics, will be dissipated when a space vehicle enters dense planetary atmospheres. The bulk of this energy is exchanged during the entry phase by converting the kinetic energy of the vehicle into thermal energy in the surrounding atmosphere through the formation of a strong bow-shock ahead of the vehicle. In general, about 90% of the energy dissipated to the atmosphere is carried away from the vehicle through convection and radiation, leaving about 10% to be absorbed back into the vehicle as thermal energy. A Thermal Protection System (TPS) is used to mitigate this heat load and ensure that the temperature limits of critical components on board are not exceeded during the entry phase.

The aerothermal environment surrounding a vehicle during atmospheric entry is extremely complex. As such, prediction of the heating rate which is experienced by the thermal protection system remains an imperfect art, leading to very large safety margins for the vehicle design. Failing to correctly predict the heat loads and associated material response of the TPS during the design phase can lead to catastrophic mission failure.

To address this problem, experimental facilities capable of generating high-speed and plasma flows are developed to study different aspects of atmospheric entry flows, together with physico-chemical models used to describe the state of the flow at a given time and conditions. Numerical methods as well are constantly improved to obtain approximate solutions of this complex system. The delicate interplay of experiments, models, and numerical methods represents the main source of knowledge about the system and its coupling mechanisms among the different physical phenomena present.

Uncertainties on the margin definition are present at every step of the design chain posing a high need for uncertainty quantification methods to analyze our results, validate our models and give us the way for improvement and research to be done.

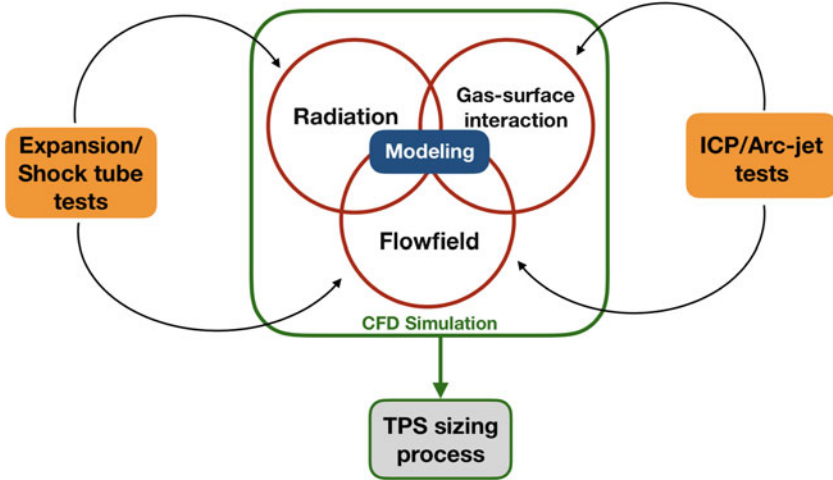


Fig. 16.1 Testing methodology for high-speed re-entry TPS sizing

16.1.1 Aerothermodynamics Testing

Ground testing is essential for the development and design of aerospace vehicles [1]. Firstly, it represents a convenient step in the testing procedure at reduced cost, compared to a real flight experiment. Secondly, ground tests allow a better control of the environment and a direct application of the measurement techniques to investigate the complexities present in high-speed flows. They also have a very practical use to assess the performance of TPS materials, to determine their surface and in-depth properties, and to characterize their high temperatures response when exposed to dissociated flows.

In the particular case of atmospheric entry flows, the different time scales related to each physical phenomenon at play force us to use a **set of facilities** to fully characterize the flow and material response. Thermal environment is duplicated in plasma facilities while radiation and non-equilibrium effects are reproduced in hypersonic wind tunnels. Figure 16.1 shows the complex intricacies present when we aim at fully evaluating the TPS performance, both numerically and experimentally.

The missing link between these facilities and the real flight conditions is the extrapolation methodology used to define the testing conditions correspondent to certain atmospheric entry conditions when all the phenomena are taken into account and coupled together.

16.2 Ground Testing in High-Enthalpy Facilities

In order to provide relevant data, the severe environment encountered by a spacecraft during re-entry has to be reproduced in ground based facilities. High-speed flows and high thermal loads must be provided simultaneously which represents a limitation for our current technical capabilities as briefly explained in the previous section [2]. Facilities able to provide high Mach number flows are characterized by limited test time, insufficient to reproduce the characteristic heat loads for thermal protection material performance testing. Plasma wind tunnels instead allow long duration heating tests, but flow Mach numbers are limited. To overcome this limitation, testing methodologies have been elaborated to simulate high temperature effects in hypersonic flows in high-enthalpy wind tunnels, like binary scaling [3–5] and the Local Heat Transfer Simulation (LHTS) [6, 7] for plasma flows. Figure 16.2 shows the different key regions that are formed during a re-entry flight, roughly, the shock and post-shock relaxation regions on one hand, and the thermal boundary layer on the other. A complete understanding of the real flight must coordinate efforts in both high enthalpy and plasma wind tunnels facilities.

The following subsections are dedicated to describe in more detail the two main types of ground testing facilities: Plasma facilities and hypersonic wind tunnels.

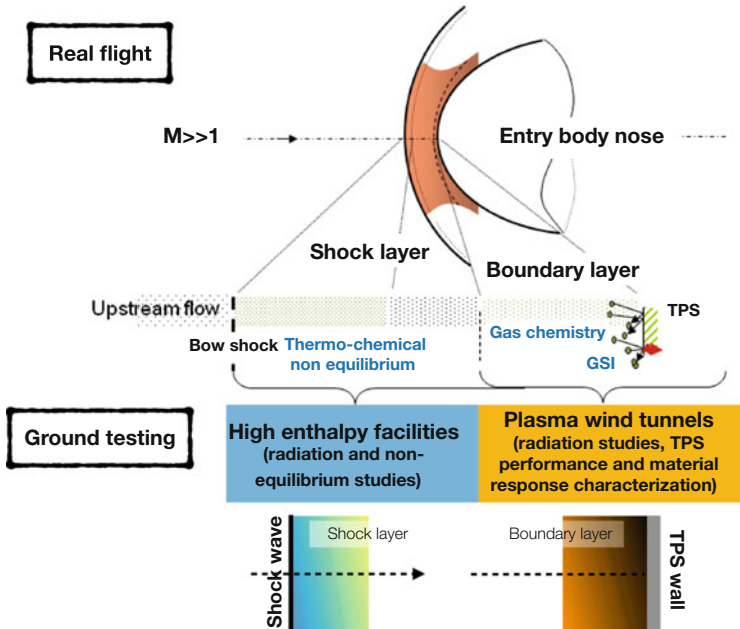


Fig. 16.2 Ground testing strategy for high-speed re-entry simulation

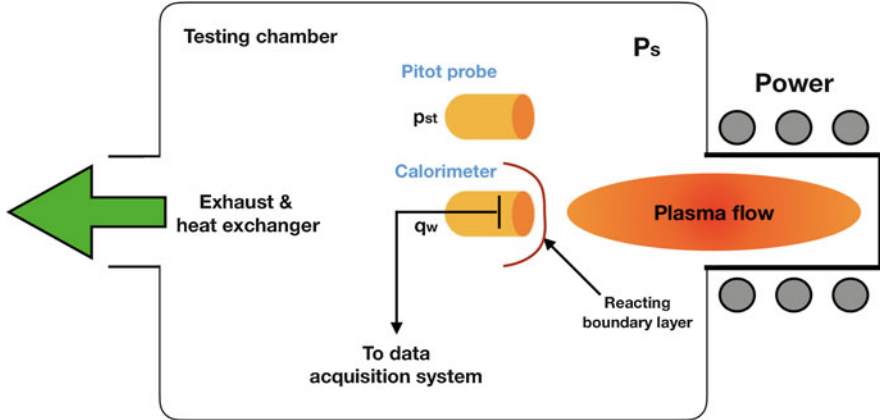


Fig. 16.3 Typical test configuration for plasma flow experiment

16.2.1 Inductively-Coupled Plasma Facilities

Inductively-coupled plasma facilities can generate a plasma discharge by electromagnetic induction. Initial ignition normally comprises the injection of argon in the torch due to its ability to generate electromagnetic discharge, which prevent free electrons to have long lifetime. An initial electrical spark, created by the strong electric field currents, introduces free electrons into the argon gas stream, which are then captured by the electromagnetic field and colliding with argon atoms, promoting further ionization. This process balances the creation of new electrons through collisions and the recombination of electrons with argon ions. The dissociated gas accelerates out of the torch in the form of a plasma jet and can be then switched to the desired test gas. Once being in equilibrium, this process can run uninterruptedly given that enough electricity, supply gas, and cooling are provided. Because the gas is heated by induction through a coil, it is creating a high purity plasma flow in the absence of eroding electrodes. In the sketch of Fig. 16.3 a common testing set-up of a plasma flow experiment is shown. In it, measurements of stagnation pressure using a Pitot probe, and of heat flux, using a calorimeter, can be taken.

The flow regime for ground testing is typically subsonic, even though supersonic conditions can be reached. Calorimetric probes are used for the determination of plasma conditions together with emission spectroscopy. Optical techniques like pyrometry, radiometry or infrared thermography are applied for the observation of the thermal protection materials response under plasma flow conditions.

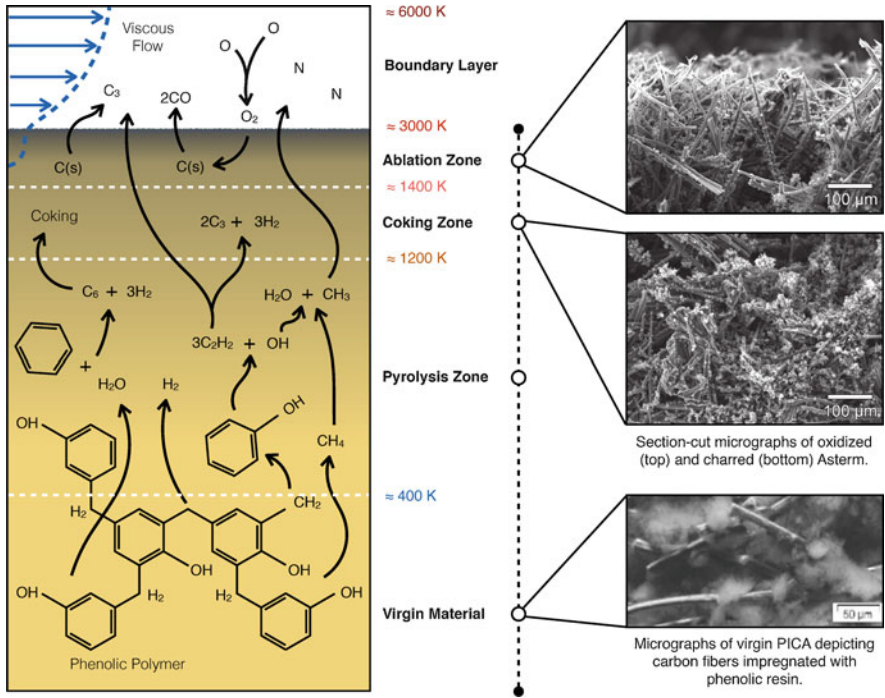


Fig. 16.4 Phenomenology of thermal protection material response. Credit: Scoggins [8]

16.2.1.1 Material Characterization

The materials used to protect planetary entry capsules from the harsh environment encountered during atmospheric entry can be classified in two main groups: reusable and ablative materials depending upon the principle they rely on to dissipate (or not absorb) the heat.

1. *Reusable heat shields* are passively cooled by re-radiating a significant amount of energy from the hot surface back into the atmosphere ($q_{rad} = \epsilon \sigma T^4$). They can survive multiple atmospheric entries at moderate orbital speed (7.9 km/s) without major changes of mass and material properties.
2. *Ablative heat shields*, in contrast, transform the thermal energy into decomposition and removal of the material. They are typically designed for one single entry, especially at velocities exceeding 10 km/s, and are generally composed of a rigid composite, reinforced with a matrix of organic resins to serve as a pyrolyzing binder, producing a char that forms on the surface followed by its ablation while the remaining solid material insulates the vehicle substructure (details are depicted in Fig. 16.4).

Different testing techniques for both types of materials are depicted hereafter.

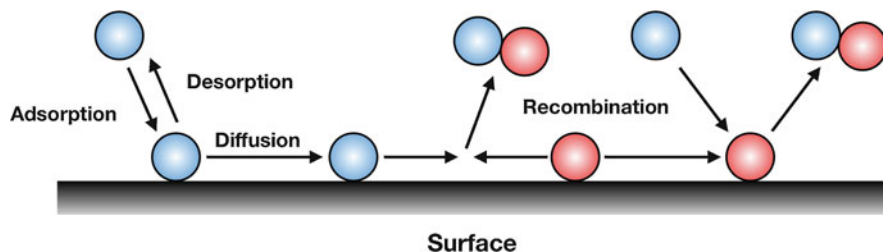


Fig. 16.5 Idealized gas-surface interactions for reusable materials

Reusable Materials

The interaction of the dissociated gas with this protection system describes the behavior of the material to act as a catalyst for recombination reactions of the atomic species in the surrounding mixture (see Fig. 16.5). Since surface reactions release a considerable quota of the heat flux to the wall one can state that the heat flux can be halved by choosing an ideal non-catalytic material [9].

A general approach to the characterization of these materials is by the determination of the so-called *catalytic recombination coefficient* (γ) [10]. This parameter can be simply defined as the ratio between the flux of recombining atoms of species i over the flux of impinging atoms on a surface. If the same catalytic parameter is assumed for every recombining specie, then a unique coefficient γ can be used.

Among the experimental procedures for the catalytic parameter determination one can distinguish between measurements taken in conditions close to the actual flight and under ideal laboratory conditions. The method used on typical induction facilities usually measures the catalytic activity on relevant flight conditions by a calorimetric method [11]. Strictly speaking, the combination of catalycity and chemical accommodation coefficient, that is the effective catalycity, is determined. Other facilities can derive the catalytic parameter through the detection of species distribution in the vicinity of TPS samples. Eventually both measurement techniques are combined with heat transfer investigations yielding a chemical accommodation coefficient.

Ablative Materials

Most research on ablative materials in the frame of the Apollo program has been carried out in arc-jet facilities, and several authors describe early experiments on various types of ablators [12–14]. They were mainly carried out in order to classify the ablative capabilities of a material to the incident heat flux, relating the surface recession rate to the surface temperature as a first step. These experiments provided a quick estimate of the amount of ablated material at steady-state recession. The averaged experimental data in use today consists of recession rates, mass loss, and

temperatures which are then used to extract thermophysical properties of the heat shield material [15]

For a comprehensive study of the properties of interest in ablators, different experimental set-ups and techniques can be used [16]:

- High-speed cameras provide a precise estimate of the recession rate of the material and can determine spectrometer probing locations.
- Infrared radiometers and pyrometers give information on the thermal behavior of the test sample by detecting the electromagnetic radiation emitted. Assuming Planck's law for black body spectral radiance, the surface emissivity and temperature can be derived providing insightful information about the surface radiometry.

Away from the sample, it is important to study the gas-phase in the boundary layer. Identification of erosion products as a function of the plasma parameters are performed by means of optical emission spectroscopy in front of the test specimen. Tracking ablation products in-situ during high-enthalpy testing offers an insight on volatilization effects at the surface and in the boundary layer. The moving surface and highly reactive boundary layer, absorbing and emitting light in multiple spectral ranges, complicate high-precision spectroscopic measurements, such as, for example, laser-induced spectroscopy. However, one can profit from the strong emission of several radiators produced by the pyrolysis gases and surface ablation for analysis of the high temperature, reacting flow, and boundary layer.

16.2.1.2 Free Stream Characterization for Validation

Validation is a vital part of any design and engineering process. Free stream conditions in ground testing facilities are used to define the flight extrapolation parameters and link experiments with their respective flight conditions. They are normally rebuilt through a combination of numerical and experimental methods resulting in their dependence upon the models used in the computational tools employed. Ground testing facilities can also provide the experimental data needed to validate the different simulation tools.

Experiments using optical emission spectroscopy are performed to determine optically and non-intrusively the temperature and composition at which the spectra detected is produced, providing the free stream conditions from experimental diagnostics when there is no probe in the testing chamber [17]. Additional data about the free stream can be obtained by using a high definition camera from an optical access window to study the fluctuations of the flow and the formation of the Mach disk in the jet during supersonic tests.

The data obtained in this manner are then processed to be used to validate the computational models for plasma flow simulations.

16.2.2 Hypersonic Wind Tunnels

Hypersonic wind tunnels represent a large group of facilities that can operate with high velocity flows ranging from Mach 5–20. A wide variety of solutions exist to provide a source of gas at sufficiently large pressure and temperature to use it as a working fluid for a hypersonic wind tunnel. Excellent reviews of the different types of hypersonic wind tunnels together with a historical perspective are given in [18, 19]. A detailed analysis of the limiting parameters of hypersonic facilities can be found in [20]

The conventional design of closed-circuit supersonic wind tunnels can hardly be extended to reach larger flow enthalpies/velocities. In order to reach the lower range of hypersonic Mach numbers, numerous wind tunnels use instead the *blowdown* principle: the test gas is stored in a reservoir at high pressure and high temperature and subsequently released through a nozzle for a test duration of few seconds up to few minutes depending on the amount of gas initially stored and the test conditions.

For Mach numbers larger than 6 or so, the blowdown technique is again rarely applicable and other ways to achieve hypersonic flows have been sought, leading to the *impulse-type* of wind tunnels. Typically, kinetic, thermal, electrical or chemical energy is stored over a long period of time with a low input power and rapidly released. In sum, hypersonic wind tunnels allow to produce high-enthalpy flows but with a reduced run time, of the order of milliseconds.

16.2.2.1 Non-equilibrium Effects

Simulating the features of non-equilibrium post-shock environment in high-enthalpy facilities for forebody stagnation regions is an important aspect of testing in hypersonic wind tunnels. Considering a blunt body at a velocity of 7 km/s, the temperature immediately after the shock is around 14,000 K, and around 8000 K downstream the shock, where the flow may return to equilibrium. At such high temperatures, thermo-chemical effects have a significant influence on the flow.

These remarks remind that the experimental hypersonic simulation for studying non-equilibrium phenomena requires an accurate understanding of the ground testing facility operation and a careful interpretation of the experimental data. To give typical reference for air at a pressure of 1 atm, vibrational excitation begins at 800 K, O_2 begins to dissociate at 2500 K and is fully dissociated for 4000 K, point for which N_2 begins to dissociate. At 9000 K, N_2 is fully dissociated and ionization begins. In the aim at reproducing high temperature flow phenomena in ground testing facilities one need to consider the conservation of mass when taking into account thermo-chemical reactions that produce and/or destroy different chemical species.

Different testing techniques are used in these facilities to gather data on several aspects of the flow around the entry body where non-equilibrium effects can take place [21]. The estimation of the free stream static pressure, for instance, is normally

derived from other measurements and often based on the assumption of an isentropic flow expansion. However, this is a severe assumption which is violated as soon as non-equilibrium effects are present in the flow due to the irreversibilities introduced. Non-equilibrium effects easily occur in hypersonic wind tunnels having stagnation temperatures large enough to excite the vibrational modes of the gas species. A dedicated free stream static pressure probe can be used to directly calibrate the free stream conditions in the chamber.

Stagnation probes are set up to obtain measurements of the heat flux and pressure that are mainly used to derive the total enthalpy of the free stream based on a series of assumptions about the flow in the facility. Optical diagnostics have been also developed for these facilities as a mean to obtain qualitative information about different flow features of interest concerning transition on the boundary layer and shock formation among others [22].

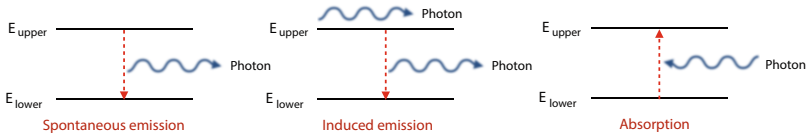
16.2.2.2 Shock Layer Radiation

In addition to dissociation and ionization, collisional excitation, which comprises the phenomenon of excitation of atoms and molecules by means of collisional energy, may lead to thermal radiation. Excited atoms and molecules can spontaneously emit photons and drop to lower energy states. The radiant energy can then be absorbed by other atoms and molecules in the flowfield, causing particles to jump to higher energy levels, or directly absorbed by the surface of the vehicle.

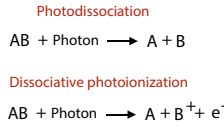
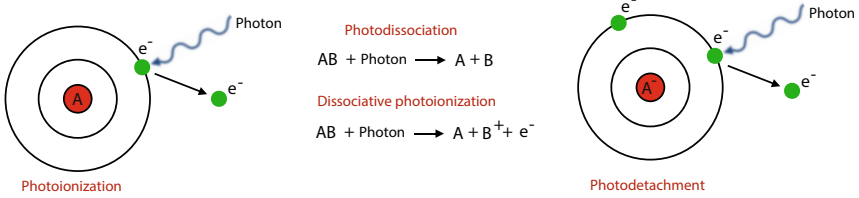
Emission and absorption from bound levels to other bound levels is referred to as *bound-bound radiation*. The emission and absorption spectra of bound-bound processes is highly oscillatory in nature due to the many transitions between discrete internal energy levels of atoms and molecules. For sufficiently energetic atoms and molecules, absorption of a photon may lead to dissociation or ionization. In particular, the photo-dissociation of O_2 and photo-ionization of N and O are common in air. These processes are called *bound-free* since the particles in question begin the process bound to one another and are separate or *free* at the end. The reverse processes are likewise termed *free-bound*. Finally, free electrons may also contribute to radiation. As an electron passes through the electric field of another charged particle, it may undergo a deceleration and emit a photon with energy equal to the difference between the kinetic energy of the electron before and after the collision. This process is known as *Bremsstrahlung* from the German words *bremsen* for *to brake* and *strahlung* for *radiation*. Bremsstrahlung is also called *free-free radiation*, since transitions occur between two unbound electrons. Bound-free and free-free processes exhibit a continuous spectrum, since the energy transitions are not limited to discrete jumps. Figure 16.6 gives illustrations of the different processes.

To experimentally assess the relative importance of these radiative processes, hypersonic wind tunnels can use optical diagnostics to detect the emission or absorption spectra in different parts of the flowfield, especially the post-shock region. The data gathered through spectrometers is used to validate radiation models

Bound-bound processes



Free-bound or bound-free processes



Free-free processes

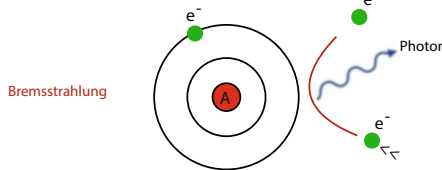


Fig. 16.6 Radiative processes in gaseous media

that are coupled with flow solvers to simulate the complex environment of the re-entry [23].

16.2.3 Aleatory Uncertainties

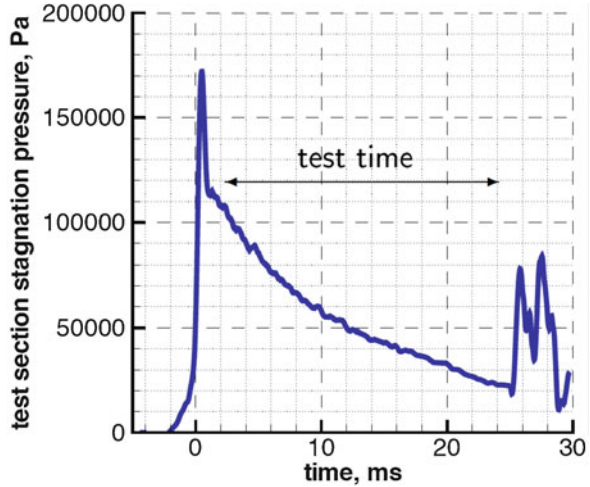
The physical processes described in this section occurring in both high velocity and high temperature laboratory flows are complex and we still struggle to obtain a full detailed description of their behavior. Experiments of these kinds are constantly shedding light on the underlying physical processes that take place at the various levels.

Learning from experiments implies knowing the reliability faced when performing measurements. Information cannot be obtained from unreliable sources as they cannot be trusted. To this end, uncertainty quantification provides a framework to analyze our data in this complex system and estimate uncertainty levels [24].

An example of the seemingly *randomness* that dominates the measurements performed in the aforementioned facilities is depicted below in Fig. 16.7.

This time-resolved measurement of stagnation pressure in the test section of a hypersonic wind tunnel fails to be a smooth curve with time, with observable wrinkles along the way. Traditionally, a probabilistic approach is used to quantify

Fig. 16.7 Hypersonic wind tunnel experimental data (Credit: G. Grossir, VKI Longshot)



the mean and deviation of the pressure values with time that can be then properly treated by uncertainty quantification codes as inputs to propagate or infer model parameters uncertainties. In this probabilistic framework, Gaussian or normal distributions are often assumed for the experimental uncertainties models in which a given measurement (one point in time) should be considered a sample from this distribution rather than the mean value itself for which the fluctuations with time will have to converge.

Another example of typical measurements obtained in plasma wind tunnels can be seen in Fig. 16.8 where the probe surface temperature is recorded with time.

The random fluctuations of the sensor, in this case a pyrometer, can be more clearly appreciated.

To this end, we can say that in the physical systems studied here, two basic kinds of uncertainty are considered: systematic, reproducible errors affecting the whole experiment, and random uncertainties associated with intrinsic variations in the experimental conditions, in the sensor readings or deficiencies in defining the quantity being measured. Not considering the statistical variability, stemming from random uncertainties, when assessing the reliability of the measurements can lead to erroneous conclusions. A more realistic approach to treat experimental uncertainties needs to be studied and developed in order to have a better characterization of our inputs and model parameters. In turn, this will produce better predictions, improving the solutions for our engineering systems.

16.3 Physico-Chemical Models and Computational Tools

Important research efforts have been conducted towards the development of numerical simulation tools for high-enthalpy flows. Physico-chemical models are con-

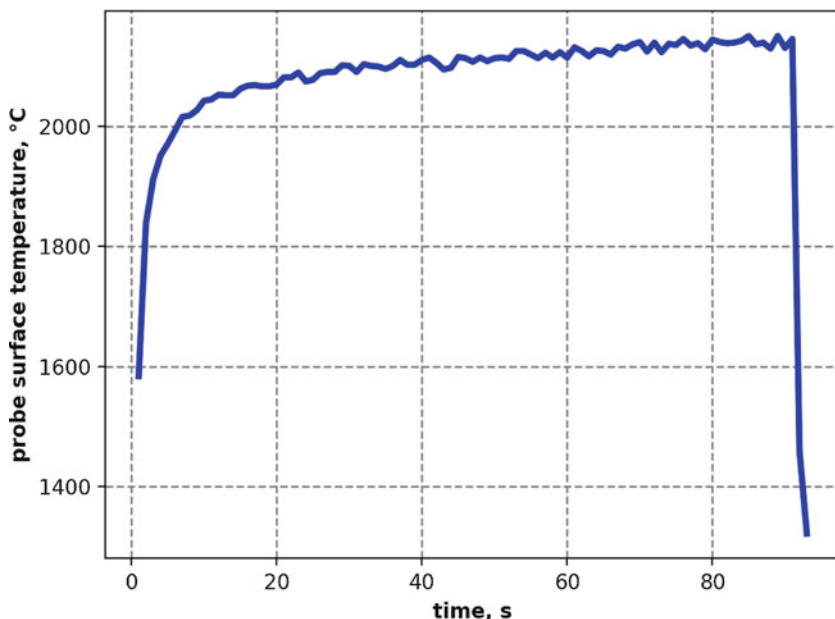


Fig. 16.8 Plasma wind tunnel experimental data (Credit: B. Helber, VKI Plasmatron)

stantly being improved and implemented into computational fluid dynamics codes, essentially dealing with Earth re-entries. In particular, thermodynamic and transport properties of Earth atmosphere as well as chemical data are normally gathered in a dedicated library. Thermo-chemical non-equilibrium effects are studied and simulated for the different experimental facilities for model validation. Gas-surface interactions and their influence on the wall heat flux are also investigated for general hypersonic re-entry flows and also chemically reacting boundary layers.

Computational methods implement these theoretical models into simulation frameworks to provide solutions and predict certain system behaviors. The reliable prediction of heat fluxes to hypersonic vehicles remains a challenge. Heat fluxes require accurate predictions of temperature gradients at the vehicle surface, which are considerably more sensitive to the surface-normal grid resolution than other quantities, such as the surface pressure. At least as important is the presence of strong shock waves to which a significant quota of the error is attributed when the grid is not shock-aligned. This is particularly obvious in stagnation regions. If the error dominates the flowfield, the shock can lean outward and develop a catastrophic breakdown of the solution.

The resolution of the temperature field and other gradients across the boundary layer and at the vehicle surface present additional challenges to the numerical methods, the near-wall grid spacing must be very tight so as to resolve key gradients. Atmospheric entry flows are also characterized by complex processes taking place within the gas and as the gas interacts with the surface. At high pressures or

relatively slow flow conditions, the thermo-chemical state adjusts rapidly to the flow state and is close to thermal and chemical equilibrium. However, in many applications, the chemical kinetics and relaxation time scales are similar to the flow motion scales, defining a state of non-equilibrium. In this state, equations for the different chemical species must be additionally solved adding complexity to the system. Boundary conditions are also complicated because of their dependence on the surface material and flow conditions, injecting ablation species into the boundary layer.

The coupling of the different mechanisms in the flow together with the difference in time scales of the processes involved makes this system an ideal place to develop and improve numerical methods that can provide accurate and fast predictions. A particular example of a computational model for atmospheric entry flows is described in Sect. 16.3.1.1.

Hereafter follows a review of the most important theoretical aspects and models used to describe the system.

16.3.1 *Governing Equations for Atmospheric Flows*

Fluid dynamics allows to describe macroscopically a plasma flow. To that aim, the Navier–Stokes equations deal with the conservation of mass, momentum, and energy in a small element of volume of plasma. Transport phenomena occur at the volume surface. First, diffusion of chemical species and heat fluxes arise through the interface. Second, shear stresses applied on the surface participate to the equilibrium of the forces acting on the volume. The transport fluxes can be interpreted at a microscopic scale by studying binary collisions among particles of the plasma. The Boltzmann transport equation is the fundamental governing equation for non-equilibrium flow systems. The link between both macro and microscales is established by means of the kinetic theory.

The dimensionless quantity Knudsen number (Kn) is quite often the criterion that distinguishes the continuum and rarefied gas regimes both present in atmospheric entry flows. The Knudsen number is defined as follows:

$$Kn = \frac{1}{vt_f} \quad (16.1)$$

where v represents the intermolecular collision rate and t_f the characteristic flow time. Fig. 16.9 shows the different numerical methods used to describe atmospheric entry flows over a wide range of Knudsen numbers.

The local Knudsen number being less than $Kn < 0.01$ defines the state of continuum flows when the average distance a molecule travels before colliding with another molecule (defined as mean free path) is smaller than a characteristic length related to the physical characteristics of the flowfield in question.

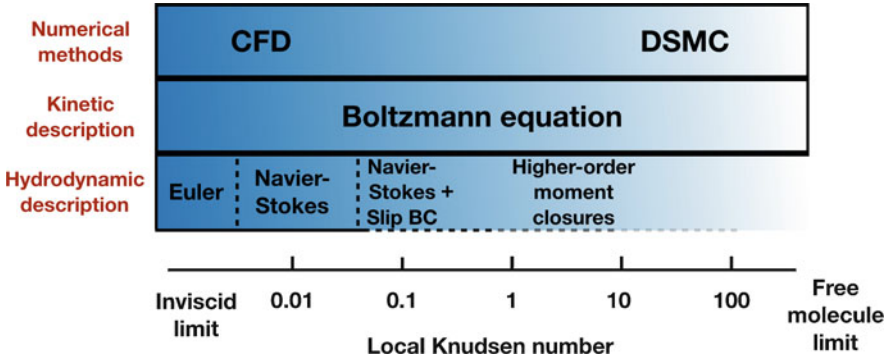


Fig. 16.9 General description of physical models used for different Knudsen number flow regimes

In the particular case of atmospheric entry flows discussed here, the continuum regime is encountered when dissipating the high kinetic energy of the high-speed, hypersonic flow along the regions of the lower atmosphere in the case of Earth re-entries. The way of dissipating this large amounts of energy is by the influence of friction within the boundary layer. The extreme viscous dissipation is responsible for the excitation of internal energy modes within molecules and to cause dissociation and ionization.

The set of conservation equations used to describe an unsteady, compressible, three-dimensional flow under viscous forces is a particularization of the *Navier-Stokes equations* and they take the following form in equilibrium conditions:

Global Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{16.2}$$

Momentum Conservation Equation

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla \cdot P - nq\mathbf{E}' - \mathbf{j} \times \mathbf{B} = 0 \tag{16.3}$$

Global Energy Conservation Equation

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\rho e \mathbf{v}) + \nabla \cdot \mathbf{q} - \mathbf{j} \cdot \mathbf{E}' + P : \nabla \mathbf{v} = 0 \tag{16.4}$$

with density ρ , stress tensor $P = \sum_{j \in S} P_j$, hydrodynamic velocity v , energy e , and heat flux \mathbf{q} under the effects of an electric field described in the hydrodynamic frame $\mathbf{E}' = \mathbf{E} + \mathbf{v} \times \mathbf{B}$ with mixture charge $q = \sum_{j \in S} x_j q_j$ and mixture conduction current $\mathbf{j} = \sum_{j \in S} \mathbf{j}_j$ (being $\mathbf{j}_i = n_i q_i \mathbf{V}_i$)

For a dissociated, ionized gas, the system is not closed. Extra equations describing the species continuity and energy conservation including a diffusion term, must be implemented to fully describe the system [25]

Species Continuity Equation

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{v}) + \nabla \cdot (\rho_i \mathbf{V}_i) = 0, \quad i \in S \quad (16.5)$$

with the additional constraint $\sum_{j \in S} \rho_j \mathbf{V}_j = 0$

Species Energy Conservation Equations

$$\frac{\partial}{\partial t} (\rho_i e_i) + \nabla \cdot (\rho_i e_i \mathbf{v}) + \nabla \cdot (\mathbf{q}_i) - \mathbf{j}_i \cdot \mathbf{E}' + \rho_i \mathbf{V}_i \cdot \frac{\partial}{\partial t} \mathbf{v} + P_i : \nabla \mathbf{v} = 0, \quad i \in S \quad (16.6)$$

Being \mathbf{V}_i the diffusion velocity of species i .

Simulating re-entry flows in inductively-coupled plasma facilities requires and extra effort to model the electromagnetic field as seen in the previous equations. In this case, the *Maxwell equations* [26] are solved to obtain the electromagnetic field, playing a role in the flowfield through the effect of the Lorentz force and Joule heating [27]. In Section 16.3.1.1, a particular computational model that implements these continuity and conservation equations for a plasma wind tunnel is introduced for its role in determining experimental conditions and other non-measured parameters of the flow field.

16.3.1.1 Resistive Magneto-Hydrodynamics (MHD) Model

In the particular case of inductively-coupled plasma facilities, the hydrodynamic and electromagnetic equations (MHD) are discretized and solved. The physical description incorporates various kinetic, chemical, thermodynamic, and mathematical tools developed in the previous chapters. The following hypothesis are assumed to be satisfied in the plasma flow:

$f \ll f_p$ The plasma frequency $f_p = n_e^{1/2} q_e / [2\pi(\epsilon_0 m_e)^{1/2}]$ represents the oscillation frequency of the free electrons about their equilibrium positions [28]. The displacement current can be neglected in the plasma provided that the torch frequency f is lower than the plasma frequency [25].

$L_C \ll \lambda_{EM}$ The current in the inductor generates electromagnetic waves of length $\lambda_{EM} = 1/[f(\epsilon_0 \mu_0)^{1/2}]$. When the coil length L_C is smaller than the electromagnetic wavelength, the displacement current in the inductor remains negligible in the computation of the electromagnetic field.

LTE Local Thermodynamic Equilibrium is a strong hypothesis. The composition of the mixture of the various species present in the plasma is considered to instantaneously adapt itself to changes in the flow, such that the plasma can be



Fig. 16.10 Inductive plasma wind tunnel: torch, coil, and probe (Credit: T. Magin [27])

thought of as a single gas of defined composition or a gas in chemical equilibrium [29]. The pressure and temperature of the gas are sufficient to uniquely define the distribution of the species, which is therefore not dependent on the history of the flow, and a single temperature can be used to describe the flow. The gas is said to be in thermal equilibrium. In this hypothesis, when the flow is considered both in chemical and thermal equilibrium, the gas is defined as being in a state of local thermodynamic equilibrium.

No Elemental Demixing The elemental demixing consists in the separation of chemical elements by diffusion. This phenomenon, encountered in situations of both chemical non-equilibrium [30, 31] and equilibrium [32]. The elemental fraction is assumed constant in the flow.

Low Re In the jet, the Reynolds number $Re = \rho v L^0 / \eta$ remains sufficiently low in the probe measurement area to avoid turbulence effects.

Low M In our applications, the Mach number $M = v/a$ is found to be low such that the flow remains subsonic. This information is important to design a suitable discretization scheme for the convective fluxes.

No Radiation and Steady State The Navier–Stokes equations are averaged in time over an oscillation period of the electromagnetic field. A steady-state solution of these equations is sought.

The geometry of an inductive plasma wind tunnel is shown in Fig. 16.10. An axisymmetric configuration is retained to model the facility. Symmetry implies that derivatives in the θ direction cancel, $\partial/\partial\theta = 0$. The inductor is represented by n_r parallel coil rings considered infinitely thin.

The coupling of the electromagnetic forces governed by the Maxwell equations to the Navier–Stokes equations which describe the flow field is realized by the Lorentz forces acting on the charged particles subject to the electromagnetic field, the Joule dissipation which heats up the plasma, and the temperature dependency of the electrical conductivity of the plasma. Figure 16.11 shows graphically the coupling between both sets of equations.

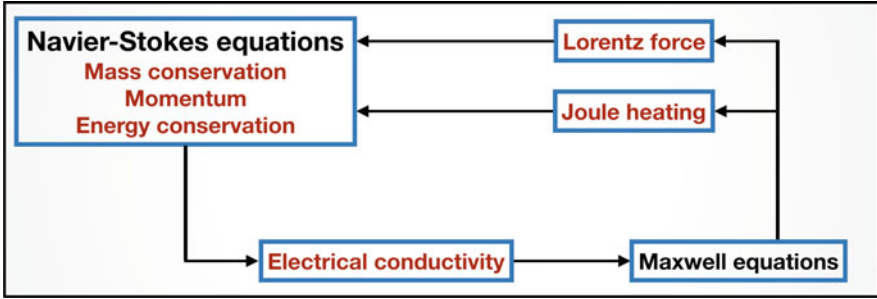


Fig. 16.11 Coupling between hydrodynamic and electromagnetic field equations

16.3.2 Closure Models

The governing equations described in the previous section require the evaluation of mixture and species thermodynamic properties, transport fluxes, chemical kinetics, and internal energy. In particular, energy and enthalpy are explicitly necessitated by the conservation of energy. Thermodynamic properties are depicted hereafter followed by the transport, chemistry, and energy transfer terms as they require the evaluation of the first in their formulation.

16.3.2.1 Thermodynamic Properties

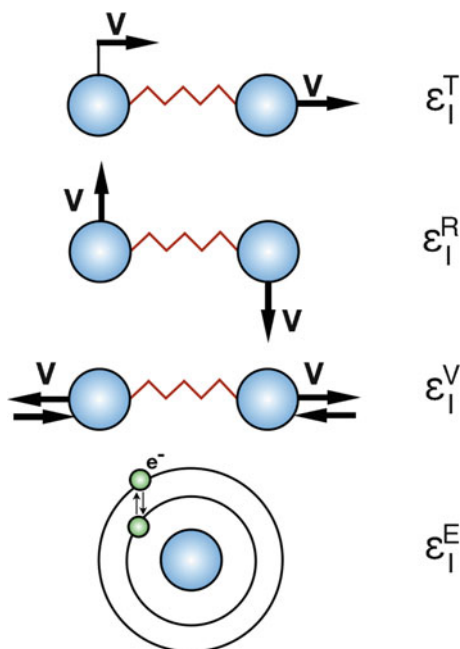
To properly describe the behavior of the flow around a re-entry vehicle, it is necessary to derive the properties of a mixture of gases, where different species coexist at a given time, that can chemically react with each other due to the high temperatures encountered in this process.

For a better understanding, we will proceed by looking at the behavior of pure gases (perfect gas consisting of a single species) and extend it to a general mixture of gases for which the models are depicted and explained.

Thermodynamics of Pure Perfect Gases

Quantum mechanics dictates that atoms and molecules are permitted only discrete energy levels [33]. For atoms, this energy is contained within translational and electronic energy modes. Molecules have two additional energy storage modes via rotation and vibration of the molecule. In general, all energy modes are coupled. For weakly interacting particles (dilute gases), however, translational energy may be considered decoupled from the internal energy. Note that this use of *internal energy* should not be confused with the typical fluid dynamics description of the internal energy of a gas, which includes translational energy. Here, internal energy

Fig. 16.12 Energy modes of a generic molecule



is meant to differentiate between energy associated with the translation of the center of mass of a particle and the energy associated with the relative motion of its constituents (nuclei and electrons). While translational energy levels are discrete, the spacing between levels is extremely small. For all practical purposes, this permits a semi-classical approach in which translational energy is assumed continuous while internal energy is left discrete. Furthermore, it is assumed that the populations of internal energy levels satisfy Maxwell–Boltzmann statistics, such that the quantum effects differentiating bosons (Bose–Einstein statistics) and fermions (Fermi–Dirac statistics) are negligible [33]. The rigid rotator and harmonic oscillator approximations for molecules are considered in the ground state. The electronic levels of atoms and molecules are selected based upon a simple cut-off criterion [27].

In a gas composed of independent particles that are identical, let us assume that the total energy of level I for a molecule is given by its translational, rotational, vibrational, and electronic contributions.

If we combine the different modes depicted in Fig. 16.12, the total energy of the molecule is obtained:

$$\epsilon_I = \epsilon_I^T + \epsilon_I^R + \epsilon_I^V + \epsilon_I^E, \quad I = 1, 2, \dots \quad (16.7)$$

The total energy of a system of identical particles N_I is

$$\sum_I N_I \epsilon_I = E \quad (16.8)$$

Statistical derivation of thermodynamic properties can be found in standard textbooks [34]. The models for the different energy modes are depicted for a mixture as a generalization of this particular case.

Mixture of Inert Perfect Gases

In a mixture of different species, the same principles apply. The equations for the thermodynamic properties of the different energy modes are depicted hereafter for the species i of the mixture:

$$e_i^T = \frac{3}{2} \frac{k_B T}{m_i} \quad (16.9)$$

$$h_i^T = e_i^T + \frac{k_B T}{m_i} \quad (16.10)$$

$$s_i^T = \frac{h_i^T}{T} + \frac{k_B}{m_i} \ln \left[\frac{k_B T}{p} \left(\frac{2\pi m_i k_B T}{h^2} \right)^{\frac{3}{2}} \right] \quad (16.11)$$

Electronic energies, enthalpies, and entropies of atoms and molecules read

$$e_{E_i} = h_{E_i} = \frac{k_B}{m_i} \frac{\sum_n g_{in} \theta_{E_{i,n}} \exp\left(\frac{-\theta_{E_{i,n}}}{T}\right)}{\sum_n g_{in} \exp\left(\frac{-\theta_{E_{i,n}}}{T}\right)} \quad (16.12)$$

$$s_{E_i} = \frac{h_{E_i}}{T} + \ln \sum_n g_{in} \exp\left(\frac{-\theta_{E_{i,n}}}{T}\right) \quad (16.13)$$

Quantity g_{in} stands for the degeneracy of the electronic level n of the species $i \in H$ and $\theta_{E_{i,n}}$ for its energy (in K unit). The number of electronic levels retained is limited for mathematical and physical standpoints [35].

For linear molecules, rotational energies, enthalpies, and entropies are expressed as:

$$e_{R_i} = h_{R_i} = \frac{k_B T}{m_i} \quad (16.14)$$

$$s_{R_i} = \frac{h_{R_i}}{T} + \frac{k_B}{m_i} \ln \left(\frac{T}{\sigma_i \theta_{R_i}} \right) \quad (16.15)$$

where θ_{R_i} stands for the rotational characteristic temperature (it is assumed that $T \gg \theta_{R_i}$). The symmetry number $\sigma_i = 1$ or 2 depending on whether the molecule is heteronuclear (CO, NO, NO_+) or homonuclear ($CO_2, N_2, O_2, N_{+2}, O_{+2}$). Diatomic molecules possess one single vibrational mode. Vibrational energies, enthalpies, and entropies read

$$e_{V_i} = h_{V_i} = \frac{k_B}{m_i} \sum_m \frac{\theta_{V_{i,m}}}{\exp \left(\frac{\theta_{V_{i,m}}}{T} \right) - 1} \quad (16.16)$$

$$s_{V_i} = \frac{h_{V_i}}{T} - \frac{k_B}{m_i} \sum_m \ln \left[1 - \exp \left(-\frac{\theta_{V_{i,m}}}{T} \right) \right] \quad (16.17)$$

where $\theta_{V_{im}}$ stands for the vibrational characteristic temperature associated to the vibrational mode m . To account for the energy released in the gas by chemical reactions between the species, a common level from which all the energies are measured is established by means of the formation enthalpy h_{F_i} at 0 K. Gathering contributions of the various degrees of freedom, species enthalpies are written:

$$h_i = h_{T_i} + h_{E_i} + h_{F_i}, \quad \text{for atoms} \quad (16.18)$$

$$h_i = h_{T_i} + h_{E_i} + h_{R_i} + h_{V_i} + h_{F_i}, \quad \text{for molecules} \quad (16.19)$$

$$h_e = h_{T_e} + h_{F_e}, \quad \text{for electrons} \quad (16.20)$$

The formation entropy at 0 K is zero, therefore species entropies read

$$s_i = s_{T_i} + s_{E_i}, \quad \text{for atoms} \quad (16.21)$$

$$s_i = s_{T_i} + s_{E_i} + s_{R_i} + s_{V_i}, \quad \text{for molecules} \quad (16.22)$$

$$s_e = s_{T_e} + \frac{k_B}{m_i} \ln 2, \quad \text{for electrons} \quad (16.23)$$

The thermodynamic properties of the mixture are readily obtained by weighting the species properties by species mass densities, $\rho h = \sum_{j \in S} \rho_j h_j$ and $e = h - p/\rho$. The contribution of the entropy of mixing is added to evaluate the mixture entropy, $\rho s = \sum_{j \in S} \rho_j s_j + k_B \sum_{j \in S} n_j \ln(1/x_j)$.

For a mixture of reacting gases where chemical reactions take place, the production and deletion of species must be taken into account in the models.

16.3.2.2 Transport Phenomena

Closure of the transport fluxes is derived through a multiscale Chapman–Enskog perturbative solution of the Boltzmann equation as explained in Sec. 16.3.1. The fluxes considered are the stress tensor, the diffusion fluxes, and the heat flux which play a major role in hypersonic flows.

Stress Tensor

$$P = p \mathbf{I} - \eta \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T \right] - \left(\kappa - \frac{2}{3} \eta \right) \nabla \cdot \mathbf{v} \mathbf{I} \quad (16.24)$$

where η and κ are the dynamic (shear) and bulk (volume) mixture viscosities. In an expansion or compression of a gas, the work done by the pressure alters immediately the translational energy of the molecules. Bulk viscosity arises from a time-lag necessary to reequilibrate the translational and internal energies through inelastic collisions. This term is nearly always neglected in hypersonic flow calculations under the assumption that $\kappa/\eta \ll 1$. However, in [36], Giovangigli et al. have shown that this assumption is not always valid for larger polyatomic gases.

The dynamic viscosity η is obtained from the first Laguerre–Sonine polynomial approximation of the Chapman–Enskog expansion [27].

Diffusion Fluxes

Species diffusion velocities can be obtained from the multicomponent diffusion coefficient matrix D_{ij} by

$$\mathbf{V}_i = - \sum_{j \in S} D_{ij} (\mathbf{d}_j + k_j^h \nabla \ln T_h + k_j^e \nabla \ln T_e) \quad (16.25)$$

where \mathbf{d}_j are the species specific driving forces defined as:

$$\mathbf{d}_j = \frac{\nabla p_j}{n k_B T_h} - \frac{y_j q}{n k_B T_h} \nabla \ln p - \kappa_j \mathbf{E} \quad (16.26)$$

with $\kappa_j = x_j q_j / k_B T_h - y_j q / k_B T_h$, q the mixture charge and \mathbf{E} the electric field. The multicomponent diffusion coefficient matrix D_{ij} is a function of the species binary collision integrals and compositions [8].

Heat Flux

The heat flux vector in Eq.(16.4) accounts for the energy transferred through diffusion, thermal diffusion, and conduction, such that:

$$\mathbf{q} = \sum_{j \in S} \rho_j h_j \mathbf{V}_j + nk_B T_h \sum_{j \in S} (k_j^h + k_j^e) \mathbf{V}_j - \sum_{m \in \mathcal{M}} \lambda_m \nabla T_m \quad (16.27)$$

where $k_{T_j}^h$ and $k_{T_j}^e$ are heavy particle and electron thermal diffusion ratios and $\lambda_m = \lambda_m^t + \lambda_m^{int}$ is the effective thermal conductivity of the global energy mode $m \in \mathcal{M}$ which can be split into translational λ_m^t and internal λ_m^{int} components. The heavy particle translational thermal conductivity is obtained from the second order Laguerre–Sonine polynomial approximation of the Chapman–Enskog expansion. Finally, the thermal conductivity associated with the internal energy of mode $m \in \mathcal{M}$, is given by the so-called Eucken corrections [37].

16.3.2.3 Chemistry and Internal Energy

Plasma flows are generally reactive: particles dissociate, recombine, and ionize. Furthermore, molecules rotate and vibrate, and the electronic configuration of atoms and molecules vary. To capture these aspects of atmospheric entry flows, the species conservation equations depicted in Sect. 16.3.1.1 have to be modified by adding additional terms to balance off the equations:

Species Continuity Equation

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot (\rho_i \mathbf{v}) + \nabla \cdot (\rho_i \mathbf{V}_i) = \dot{w}_i, \quad i \in S \quad (16.28)$$

Species Energy Conservation Equation

$$\frac{\partial}{\partial t} (\rho_i e_i) + \nabla \cdot (\rho_i e_i \mathbf{v}) + \nabla \cdot (\mathbf{q}_i) - \mathbf{j}_i \cdot \mathbf{E}' + \rho_i \mathbf{V}_i \cdot \frac{\partial}{\partial t} \mathbf{v} + P_i : \nabla \mathbf{v} = \Delta E_i, \quad i \in S \quad (16.29)$$

The chemical production source term, \dot{w}_i , found in Eq.(16.28) accounts for the production and destruction of individual species through elementary chemical reactions. A rigorous derivation of the chemical production rate from kinetic theory yields an expression of the form:

$$\dot{w}_i = \dot{w}_i^0 + \dot{w}_i^1 \quad (16.30)$$

where \dot{w}_i^0 is the zero-order Maxwellian production rate and \dot{w}_i^1 is a first-order perturbation [38].

For the energy exchange rate, different models must be adopted for each individual internal energy mode. In general, energy transfer mechanisms fall into two categories: energy relaxation processes and chemical energy exchange processes. Energy relaxation is the process in which two distributions of energy states exchange energy through elastic and inelastic collisions and *relax* to a final equilibrium distribution: Vibration-translation, free electron-vibration, vibration-vibration, elastic energy exchange between free electrons and heavy particles, etc. Chemical energy exchange processes result from reactive collisions between particles in which energy is transferred in order to promote the reaction. Important coupling mechanisms and energy relaxation processes are described in detail in [8].

16.3.3 Radiative Heating: A Coupled Phenomenon

Each radiative mechanism discussed in the previous section contributes to the net emission and absorption of photons. The energy carried by a photon at wavenumber σ is given by $hc\sigma$. Therefore, emission and absorption of photons results in a net energy transfer between points in the flowfield or from the flowfield to the vehicle surface, heating it up.

The radiant intensity $I_\sigma(\mathbf{x}, \hat{s})$ is defined as the photon energy flux per unit area, per elementary solid angle, per wavenumber, in the direction \hat{s} at wavenumber σ and position \mathbf{x} . In the absence of scattering, the steady Radiative Transport Equation (RTE) describes the variation of spectral intensity I_σ along a ray with length parameter s as:

$$\frac{dI_\sigma}{ds} = \hat{s} \cdot \nabla I_\sigma(\hat{s}) = \eta_\sigma - \kappa_\sigma I_\sigma(\hat{s}) \quad (16.31)$$

where η_σ and κ_σ are the local emission and absorption coefficients at point s along the ray.

Photons may be emitted in any direction. Therefore, the spectral emission coefficient is independent of direction and represents the total photon energy emitted per volume, per time, per wavenumber, and per elementary solid angle. The absorption coefficient represents the fraction of photon energy absorbed by the gas over a unit distance, and is independent of direction as well. In general, these coefficients are functions of the local energy level populations of the participating gaseous species. More details about boundary conditions for this equation and modeling can be found in [39].

Once the intensity field is known, the solution must be coupled back to the mass and energy transport equations previously depicted via the radiative surface heat flux, power, and species mass production rates due to photochemical processes.

16.3.4 Gas-Surface Interactions

As with gas-phase chemistry mechanisms describe previously through the term \dot{w}_j , heterogeneous chemistry plays an important role in the description of the thermochemical environment surrounding hypersonic vehicles. In particular, an accurate prediction of the heat flux to the surface of a vehicle may depend strongly on the correct solution of catalysis and ablation processes occurring at the gas-surface interface, known in general as *gas-surface interactions* (GSI). These processes may strongly affect the composition of the boundary layer, which in turn may alter the convective or radiative heating at or near the vehicle surface. For Martian entries, these effects have also been shown to be important downstream as ablated species radiate and increase the radiative flux to the back-shell of the vehicle. Typically, GSI models are implemented as boundary conditions along the vehicle surface.

For non-ablating thermal protection systems, catalysis may play an important role in the heating to the surface of the TPS. In particular, for Earth entries, catalytic recombination of the N_2 , O_2 , and NO at the surface are well known. Catalytic reactions do not participate in the surface mass balance but can promote substantial heat transfer. As an example, the reactions described above release approximately 950, 500, and 630 kJ per mole of product, respectively.

Surface participating reactions involve both heat and mass transfer between the surface and surrounding gas. As such, these reactions always include reactants originating from the TPS material. Examples include the nitridation or oxidation of solid carbon $C(s)$, the passive and active oxidation of silicon carbide, and the sublimation of solid carbon to form C , C_2 , or C_3 . In general, there are two types of models which are used to describe surface chemistry:

Specified Reaction Efficiency Models Perhaps the most widely used GSI model in hypersonic for its easy implementation. These models describe the mass production rate of a species at the surface as the product of the mass flux of the reactant impinging on the surface Γ_i and a reaction efficiency γ_r , which takes a value between 0 and 1. This model is especially relevant for reusable material protection systems [30].

Finite-Rate Models The model presented above assumes that gas-surface reactions occur in a single step, however, in reality these reactions are the result of multiple processes which occur at finite rates. In general, surface reactions are only allowed at a finite number of *active sites* on the surface. These active sites are highly dependent on the topology and chemical structure of the surface and are reaction dependent (see Fig. 16.5). Finite-rate surface chemistry models describe the net surface reactions as a series of the elementary processes [40, 41].

Each of these processes may occur at different temperature dependent rates. Furthermore, the reverse processes are related to the forward reaction through an equilibrium constant. The heat flux to the surface of a material is then related to the rate of change in the density of active sites. Thus, a finite-rate surface chemistry model ensures the conservation of mass and energy at the surface.

16.3.5 *Epistemic Uncertainties*

Epistemic uncertainty (also known as reducible uncertainty or incertitude) is defined as a potential deficiency that is due to a lack of knowledge [42]. It can arise from assumptions introduced in the derivation of the mathematical model 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, further inference from experimental observations and improvement of the physical models. Epistemic uncertainties can be incorporated in a Bayesian framework where a calibration of the model parameters can be carried out. The resulting uncertainties tell us what should be improved in the system to have significant reliability in our results and how we can achieve that through experiments by looking at the optimal testing conditions. This calibration of model parameters represents an important step towards a more unified framework that incorporates simulation results and experiments and their best possible reliability by giving us confidence in our results. Typical examples of sources of epistemic uncertainties in hypersonic flows are surrogate chemical kinetics models and the use of γ models for catalytic effects.

It is important to notice that in most cases the epistemic uncertainty is not well characterized by probabilistic approaches. The reason is that it might be difficult to infer any statistical information due to the **nominal** lack of knowledge, i.e. the suitability of the model that is being used to calibrate the parameters is not properly known. To cope with this, other frameworks are put in place. Simulations using different plausible models are evaluated and the largest possible confidence margins are defined.

When we explore the differences between measurements uncertainties and epistemic uncertainties, we have to remark that the former are regarded as frequentist, meaning that repetitions of a certain measurement give you a measure of the randomness of the measurement process (producing a Probability Distribution Function (PDF) that can be used in our tools). Epistemic uncertainties do not have a basis to define their PDFs. One way of tackling this issue is by adopting a Bayesian framework, as explained before, in which with sufficient prior knowledge and data, a PDF can be assigned to the various parameters inferred in this manner in our model [43].

16.4 **Putting It All Together: Extrapolation to Flight**

In the design phase of aerospace vehicles some questions need to be addressed experimentally in order to have a more accurate representation of the reality encountered during flight. As scaled-down models cannot reproduce all aspects of atmospheric entry flight, extrapolation methodologies exist for both types of facilities reviewed in Sect. 16.2.

The end part of all the design process is when all the experiments and simulations converge into the actual flight conditions and we are able to put safety margins to what has been tested and simulated to express our confidence in our tools. What follows is a review of the most widely used methodology for extrapolation from plasma wind tunnels and an account of the uncertainties we face when performing such extrapolation to flight.

16.4.1 Local Heat Transfer Simulation Methodology

When facing a design challenge, engineers look at critical values that constrain the parameters considered for the particular subsystem. When we look at the thermal protection system for space vehicles, one of the most critical and important parameters is the type of thermal protection materials we are going to use and (in case of ablative materials) the thickness. These design parameters are intrinsically associated to the heat flux absorbed by the vehicle surface while on the entry trajectory.

To address this problem, one must identify the highest possible heat flux to be encountered along the particular entry trajectory and use it as the limit to be withstood. On the different points of the vehicle surface, one can identify the stagnation region as the most likely part to suffer from the highest heat loads. It is because of this that we aim at reproducing in the wind tunnel the same heat flux on the stagnation point of our vehicle than that of the flight.

Following this line of thought, it is time to introduce the Local Heat Transfer Simulation (LHTS) methodology depicted in Fig. 16.13

According to the literature in [9, 44–46], the heat flux equation can then be generalized as:

$$q_w = A Pr^{-2/3} (\rho_w \mu_w)^B (\rho_e \mu_e)^C \sqrt{\rho_e \mu_e \beta_e} (H_e - h_w) = f(H_e, \rho_e, \beta_e, h_w) \quad (16.32)$$

A direct consequence of this fact is that the ground testing of the heat flux is possible without a complete reproduction of the flight conditions. Indeed, as the enthalpy, pressure and density are directly related, Kolesnikov [6] selects only three different parameters to be matched in the ground testing facilities and in flight in order to have the same stagnation point heat flux. These parameters are the total enthalpy, total pressure, and velocity gradient at the edge of the boundary layer of the vehicle/probe.

$$H_e^t = H_e^f \quad p_e^t = p_e^f \quad \beta_e^t = \beta_e^f \quad (16.33)$$

where β denotes the velocity gradient $\partial u / \partial x$ as a measure of the velocity deflection when adapting aerodynamically to the vehicle shape with u and x as the transversal component to the stagnation line of the velocity and space, respectively.

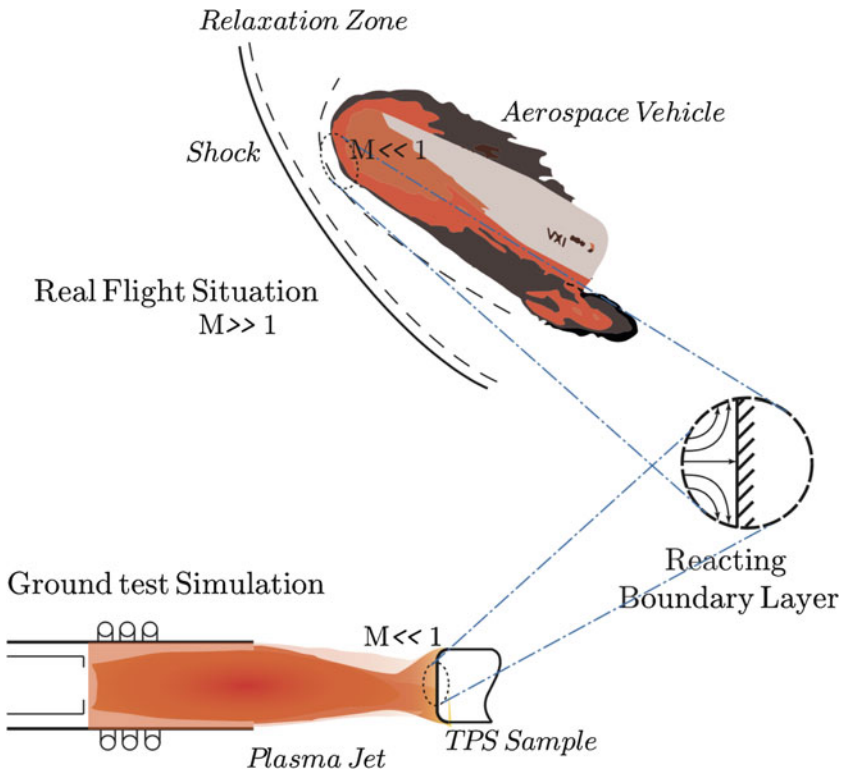


Fig. 16.13 Concept of the local heat transfer simulation (Credit: P. Solano)

One can notice that four parameters are needed in order to compute the wall heat flux according to the generalized expression depicted before (Eq. 16.32). The wall enthalpy, h_w , has a second order effect on the heat flux and should only be taken into account in the form of wall temperature if catalytic effects are unknown. According to [9], this parameter is a function of the rest of the parameters when catalysis effects are known (γ is characterized). On top of all that, it is widely accepted that $H_e \gg h_w$ as an argument to not consider this fourth parameter in the flight extrapolation problem.

16.4.2 Flight Extrapolation Uncertainties

The physico-chemical models used to describe different phenomena involved in atmospheric entry flows: radiation, flowfield, electromagnetic field, and gas-surface interactions among others, must undergo a validation campaign against experimental data. Due to the nature of this problem, experimental data can be gathered separately for each phenomena depending on the facility dedicated to the

study as seen in Sect. 16.2. According to this, each aspect of the problem has to be validated separately on ground testing facilities with occasional access to full flight data if the budget allows, but even in this situation the data is quite sparse to be validated against.

A question remains unanswered: how to identify and treat the uncertainties concerning the coupling involved in real flight when each model for each phenomenon has been validated separately in dedicated facilities?

Uncertainty quantification studies could help understand the intricacies of the different models and how this reflects on real-scale cases when put it all together in an extrapolation framework as seen in Sect. 16.4.1.

The interplay between the physical phenomena studied in the two types of facilities has to be assessed when referring to three different parameters to be reproduced on ground and in flight in such a way that the uncertainty of taking into account the correlated effects is far lower than otherwise. By doing so, we can approach the design problem with confidence intervals in the domain of the flight conditions as desired by the engineers as our final objective.

16.5 Conclusions and Remarks

Numerical simulations and experiments of atmospheric entry flows pose a challenge full of complexities.

Ground testing of entry vehicles cannot be achieved in a single facility and different aspects of the atmospheric entry flight have to be tested in dedicated facilities allowing different types of flow and testing times.

The mathematical description of the system, the physico-chemical models, reveals traces of non-linearities, coupling phenomena, and complex mechanisms whose knowledge still rests upon experimental correlation. High-speed and high temperature effects turn an otherwise well-characterized system, into a choice of models and descriptions. It is this lack of knowledge that leads to urge the community to undertake uncertainty quantification analyses for design and safety purposes.

The numerical methods depicted here clear their way through the complexities and stiffness of the equations to find approximate solutions. Uncertainty quantification must deal with the different aspects that make these flows so complex:

- **Intricate multi-physics and non-linearities** of the system when coupling flow, radiation and ablation in the same problem.
- **High dimensionality of stochastic space** when chemical reactions have to be taken into account. Each reaction is characterized through reaction constants and additional parameters to account for the production/deletion of species.
- **The computational cost** of these CFD solutions is high. The computational domain with refined mesh close to the boundaries together with possible non-equilibrium states where equations for the different species must be additionally solved are some of the issues slowing down the convergence.

- **Non-smooth solutions** in physical space imposing a shock-capturing numerical scheme.
- **Sparse experimental data** making more difficult a probabilistic treatment of the uncertainties while urging the numerical methods to be more refined in the solutions.

16.5.1 Current Margin Policies: Where Are We?

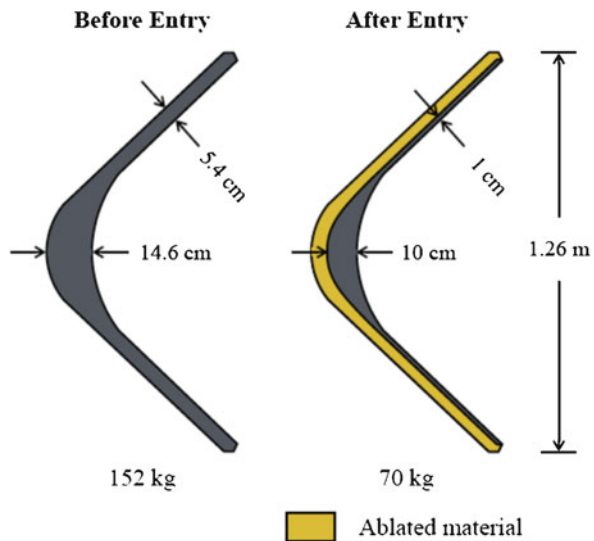
Models, particularly in aerospace sciences and material response, have largely undefined uncertainty levels for many problems (limited validation). Without well-defined uncertainty levels, it is difficult to assess the system risk and to trade risk with other subsystems. The result is typically (but not automatically) **overdesign**.

Safety factors for entry vehicles rank among the highest ones in engineering systems with an average value of 2. In atmospheric entry systems, the most critical design parameter, concerning the high heat loads encountered, is the thermal protection system thickness which directly relates to the vehicle weight. Under these premises, an over-estimation of the protection material thickness leads to higher constraints on payload weights or in a much more expensive vehicle to fly.

The opposite side of the spectrum is the underestimation of safety margins and the exposure of the crew to fatal disaster or to the robotic mission to a catastrophic end.

An example of the importance of these margins is the Galileo probe sent to Jupiter to enter the Jovian atmosphere (Fig. 16.14).

Fig. 16.14 NASA Galileo probe heatshield (Dec 1995)



The stagnation point material recession was less than predicted but the recession in the afterbody region was much higher than expected. In this case, the physical phenomena was not well-captured by the simulation tools.

The policy of safety margins must be replaced by state-of-the-art uncertainty quantification techniques for predictive engineering with benefits for the design without compromising the safety of the mission.

Acknowledgments The authors thank the European Commission H2020 programme, through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement number 722734.

References

1. O. Chazot, F. Panerai, High enthalpy facilities and plasma wind tunnels for aerothermodynamics ground testing, in *Nonequilibrium Hypersonic Flows*, ed. by E. Josyula (ARLF, 2015)
2. R.D. Neumann, Experimental methods for hypersonics: capabilities and limitations, in *Second Joint Europe-US Short Course on Hypersonic: GAMNI-SMAI and University of Texas at Austin* (USAF Academy, Colorado, 1989)
3. N.C. Freeman, Non equilibrium flow of an ideal dissociating gas. *J. Fluid Mech.* **4**(4), 407–442 (1958)
4. W.E. Gibson, P.V. Marrone, A similitude for non-equilibrium phenomena in hypersonic flight. in *The High Temperature Aspect of Hypersonic Flow, AGARDograph 69*, ed. by W.C. Nelson (Pergamon Press, Oxford, 1964)
5. G.R. Inger, C. Higgins, R. Morgan, Generalized nonequilibrium binary scaling for shock standoff on hypersonic blunt bodies. *J. Thermophys. Heat Transf.* **17**(1), 126–128 2003
6. A.F. Kolesnikov, Extrapolation from high enthalpy tests to flight based on the concept of local heat transfer simulation, in *RTO Educational Notes on Measurement Techniques for High Enthalpy and Plasma Flows* (Rhode-St-Genese 1999)
7. P.F. Barbante, O. Chazot, Flight extrapolation of plasma wind tunnel stagnation region flowfield. *J. Thermophys. Heat Transf.* **20**(3), 493–499 (2006)
8. J.B. Scoggins, Development of Numerical Methods and Study of Coupled Flow, Radiation, and Ablation Phenomena for Atmospheric Entry. Ph.D. thesis, CentraleSupélec/VKI, 2017
9. R. Goulard, On catalytic recombination rates in hypersonic stagnation heat transfer. *Jet Propulsion* **28**(11), 737–745 (1958)
10. F. Panerai, Aerothermochemistry Characterization of Thermal Protection Systems. Ph.D. thesis, Università degli Studi di Perugia/VKI, 2012
11. A. Viladegut, O. Chazot, Enthalpy characterization and assessment of copper catalysis determination in inductively coupled plasma facility, in *45th AIAA Thermophysics Conference, AIAA AVIATION Forum, (AIAA 2015-3107)* (2015)
12. R.K. Crouch, G.D. Walberg, An Investigation of Ablation Behavior of Avcoat 5026/39M Over a Wide Range of Thermal Environments. Technical Memorandum TM X-1778, NASA, 1969. 16, 21, 36
13. B.H. Wick, Ablation characteristics and their evaluation by means of arc jets and arc radiation sources, in *Secenth International Aeronautical Congress* (Paris, 1965)
14. N.K. Hiester, C.F. Clark, Comparative Evaluation of Ablating Materials in Arc Plasma Jets. NASA Contractor Report CR-1207, Stanford Research Institute/NASA, 1968
15. M.A. Covington, J.M. Heinemann, H.E. Goldstein, Y.K. Chen, I. Terrazas-Salinas, J.A. Balboni, J. Olejniczak, E.R. Martinez, Performance of a low density ablative heat shield material. *J. Space Craft Rock.* **45**(4), 854–864 (2008)

16. B. Helber, Material Response Characterization of Low-density Ablators in Atmospheric Entry Plasmas. Ph.D. thesis, VUB/VKI, 2016
17. D. Le Quang, Spectroscopic Measurements of Sub- and Supersonic Plasma Flows for the Investigation of Atmospheric Re-Entry Shock Layer Radiation. Ph.D. thesis, Universite Blaise Pascal/VKI, 2014
18. J. Lukasiewicz, *Experimental Methods of Hypersonics* (Marcel Dekker, 1973), pp. 14–16
19. C. Tropea, A.L. Yarin, J.F. Foss, *Handbook of Experimental Fluid Mechanics* (Springer, Berlin, 2007), pp. 15, 18, 19, 212, 329, and 331
20. V.K. Smith, Hypersonic overview, methodology of hypersonic testing, in *VKI Lecture Series 1993-03* (von Karman Institute for Fluid Dynamics, 1993) p. 15
21. G. Grossir, Longshot Hypersonic Wind Tunnel Flow Characterization and Boundary Layer Stability Investigations. Ph.D. thesis, ULB/VKI, 2015
22. T.J. McIntyre, H. Kleine, A.F.P. Houwing, Optical imaging techniques for hypersonic impulse facilities. *Aeronaut. J.* **111**(1115), 1–16 (2007)
23. J.B. Scoggins, L. Soucasse, P. Rivière, A. Sufiani, T. Magin, Coupled flow, radiation, and ablation simulations of atmospheric entry vehicles using the hybrid statistical narrow band model, in *45th AIAA Thermophysics Conference, AIAA AVIATION Forum, (AIAA 2015-3112)* (2015)
24. H.N. Najm, Uncertainty Quantification and polynomial chaos techniques in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **41**, 35–52 (2009)
25. M. Mitchner, C.H. Kruger, *Partially Ionized Gases* (Wiley, New York, 1973)
26. G.W. Sutton, A. Sherman, *Engineering Magnetohydrodynamics* (McGraw-Hill, New York, 1965)
27. T. Magin, A Model for Inductive Plasma Wind Tunnels. Ph.D. thesis, ULB/VKI, 2004
28. R.P. Feynman, R.B. Leighton, M. Sands, *The Feynman Lectures on Physics*, vol. II. (Addison-Wesley, Reading, 1977)
29. S. Lopes, Numerical Approach in the Design of a Plasma Reactor for Nanoparticle Production. VKI PR2008-13, 2008
30. P.F. Barbante, Accurate and Efficient Modelling of High Temperature Nonequilibrium Air Flows. Ph.D. thesis, ULB/VKI, 2001
31. D. Vanden Abeele, G. Degrez, Efficient computational model for inductive plasma flows. *AIAA J.* **38**(2), 234–242 (2000)
32. P. Rini, G. Degrez, D. Vanden Abeele, Elemental demixing in inductively coupled air plasmas at high pressures, in *37th Thermophysics Conference* Portland, 2004, AIAA 2004-2472
33. W.G. Vincenti, C.H. Kruger, *Introduction to Physical Gas Dynamics* (Krieger Publishing, Malabar, 1975)
34. J.E. Mayer, M.G. Mayer, *Statistical Mechanics* (Wiley, New York, 1946)
35. B. Bottin, D. Vanden Abeele, M. Carbonaro, G. Degrez, G.S.R. Sarma, Thermodynamic and transport properties for inductive plasma modeling. *J. Thermophys. Heat Transfer* **13**(3), 343–350 (1999)
36. V. Giovangigli, B. Graille, T. Magin, M. Massot, Multicomponent transport in weakly ionized mixtures. *Plasma Sources Sci. Technol.* **19**(3), 034002 (2010)
37. E. Eucken, Ueber das Wärmeleitvermögen, die spezifische wärme und die innere reibung der gase. *Phys. Z.* **14**, 324–332 (1913)
38. V. Giovangigli, B. Graille, Kinetic theory of partially ionized reactive gas mixtures. *Phys. A Statistical Mech. Appl.* **327**(3–4), 313–348 (2003)
39. R.M. Goody, Y.L. Yung, *Atmospheric Radiation: Theoretical Basis*, 2nd edn. (Oxford University Press, Oxford, 1997)
40. J. Marschall, M. MacLean, P.E. Norman, T.E. Schwartzentruber, Surface chemistry in non-equilibrium flows, in *Hypersonic Nonequilibrium Flows: Fundamentals and Recent Advances*, vol. 247. *Progress in Astronautics and Aeronautics* (American Institute of Aeronautics and Astronautics, Reston, 2015), pp. 239–327
41. J. Marschall, M. MacLean, *Finite-Rate Chemistry Model, I: Formulation and Reaction System Examples*, AIAA 2011-3783 (2011)

42. E.T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, 2003)
43. D.S. Sivia, *Data Analysis: A Bayesian Tutorial* (Oxford Science Publications, Oxford, 1996)
44. E.R. van Driest, *The Problem of the Aerodynamic Heating* (Aeronautical Engineering Review, 1956), pp. 26–41
45. L. Lees, Laminar heat transfer over blunt-nosed bodies at hypersonic flight speeds. *J. Jet Propulsion* **26**(4), 259–269 (1956)
46. J.A. Fay, F.R. Riddell, Theory of stagnation point heat transfer in dissociated air. *J. Aerosol Sci.* **25**(2), 73–85 (1958)

Chapter 17

Introduction to Evidence-Based Robust Optimisation



Gianluca Filippi and Massimiliano Vasile

Abstract This chapter introduces the concept of Evidence-Based Robust Optimisation (EBRO) and a few computational methods that allow calculating a robust solution when uncertainty is modelled with Dempster–Shafer Theory of evidence (DST). The chapter provides the basic elements of DST and the framework in which DST can be introduced in the robust optimisation of engineering systems. The main interest is in using DST to quantify extreme cases of epistemic uncertainty. EBRO inserts DST within an optimisation loop to generate a solution that maximises a given performance index (the quantity of interest) and the belief in its value at the same time. The chapter introduces also a decomposition approach that allows one to calculate an approximation to Belief and Plausibility in polynomial time.

Keywords Robust optimisation · Epistemic uncertainty · Evidence theory · Network model · Efficient approximation

17.1 Introduction

In the early phase of the design of an engineering system, there is a degree of uncertainty on its parts and configurations. This uncertainty is often epistemic in nature and translates into an uncertainty in the performance of the system as a whole. In order to account for this, it is common practice to add margins at system and sub-system level. This way of proceeding is commonly called *margin approach*. The margin approach evaluates the quantity of interest (for example, the mass) associated to a proposed nominal design solution, called Best Estimate (BE), and adds to it, and to each sub-system quantity of interest, a margin often called the *contingency* or *safety factor*. The safety factor accounts for the expected variations of all uncertain components. For example, a margin is added to the power demand

G. Filippi (✉) · M. Vasile
University of Strathclyde, Glasgow, UK
e-mail: g.filippi@strath.ac.uk; massimiliano.vasile@strath.ac.uk

when sizing the solar arrays on a spacecraft. The value of the quantity of interest at system level, after margins are applied, is often called the Maximum Expected value (ME). The difference between the ME and the Maximum Possible value (MP) is generally considered to be a further margin that accounts for the unexpected variation of all uncertain components [1–3]. These traditional methods, however, lack an appropriate quantification of uncertainty. As a consequence, there can be an overestimation or an underestimation of the effect of uncertainty which can lead to either an increase in costs and development time or to the occurrence of undesirable events. As it was recognised during the Columbia Accident Investigation Board (CAIB) [4], the classic pattern that brings to failure, common to many other tragic accidents [5], is the combination of production pressure, that pushes to reduce the safety margins, and a fragmented problem solving that lacks a system level understanding. Hence one can argue that there is the need to incorporate a proper quantification of uncertainty within any systems engineering approach [6, 7].

In this direction, the research field based on the synergy of optimisation algorithms and uncertainty quantification theories that is known as Optimisation Under uncertainty (OUU) [8], Uncertainty-Based Design (UBD) [9], etc. is of increasing importance and interest. In particular, it is used to quantify and optimise different uncertainty measures: robustness, reliability and in the last years also resilience. They have brought, respectively, to the approaches of Robust Design Optimisation (RDO) [10], Reliability Based Design Optimisation (RBDO) [11] and Resilience Based Design Optimisation (ReBDO) [12, 13]. Moreover, due to the complexity of engineered systems [14, 15], a further important step is given by the application of holistic system-level view and multidisciplinary principles within UBD. Uncertainty-Based Multidisciplinary Design Optimisation (UMDO) theory [16] covers for this research direction.

If one looks at the different types of uncertainty that a system can be subject to, two macro-categories can be identified: *aleatory uncertainty* and *epistemic uncertainty* [17]. Aleatory uncertainty is natural randomness which cannot be reduced. Epistemic uncertainty is due to the lack of information or incomplete data. This type of uncertainty is reducible by acquiring more knowledge on the problem.

This chapter introduces a methodology to account for epistemic uncertainty in the design for robustness of complex engineering systems. The concept proposed in this chapter is called Evidence-Based Robust Optimisation (EBRO). In EBRO, epistemic uncertainty is modelled with Dempster–Shafer Theory of Evidence (DST) [18, 19]. DST offers a natural way to assign degrees of belief to the expected performance of a system and to rigorously quantify the impact of epistemic uncertainty on the associated quantities of interest. In EBRO the system design is then optimised to maximise performance under epistemic uncertainty.

As generally recognised [16] most of the effort in UMDO community should be spent in developing complexity-reduction methods to be applied within the optimisation strategies. This is particularly true when non-probabilistic uncertainty theories, for example, DST, are adopted. Some approaches for the reduction of computational complexity of UQ with DST can be found in [20–22]. Recent approaches of UMDO with the use of DST have been proposed in [23–25]. In

particular, the work in [23], that is focused on RBDO, suggests a sequential method that incorporates mixed aleatory-epistemic uncertainty. The computational cost remains, however, exponentially complex with the problem dimension. Other works, as [25], make the assumptions of normally distributed epistemic parameters. [24] also could be intractable with high order problems.

The chapter introduces different formulations of Evidence-Based Robust Optimisation both with single and multiple objectives. It then proceeds with some techniques to efficiently compute robust solutions and with the definition of a particular model, presented in Sect. 17.4.4.1, called Evidence Network Model (ENM), that translates a complex engineering system into a non-directed graph. In this graph each node represents a sub-system or component and the associated uncertainty is quantified with basic belief assignments. Exploiting the ENM properties, a decomposition procedure is explained that reduces the computational cost to a polynomial complexity with respect to the number of interacting components.

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

17.1.1 A Classification of Uncertainty

Uncertainty comes in different forms and the nature of uncertainty suggests how it should be modelled and treated. It is, therefore, useful to classify the types of uncertainty:

- **Aleatory** uncertainties are non-reducible uncertainties that depend on the very nature of the phenomenon under investigation. They can generally be captured by well-defined probability distributions as one can apply a frequentist approach, e.g. measurement errors.
- **Epistemic** uncertainties are reducible uncertainties and are due to a lack of knowledge. Generally they cannot be quantified with a well-defined probability distribution and a more subjectivist approach is required. Two classes: a lack of knowledge on the distribution of the stochastic variables or a lack of knowledge of the model used to represent the phenomenon under investigation.
- **Structural** (or model) uncertainty is a form of epistemic uncertainty on our ability to correctly model natural phenomena, systems or processes. If we accept that the only exact model of Nature is Nature itself, we also need to accept that every mathematical model is incomplete. One can then use an incomplete (and often much simpler and tractable) model and account for the missing components through some model uncertainty.
- **Experimental** uncertainty is generally aleatory but if one consider the uncertainty associated to measurements it can be considered epistemic as it incorporates the possible lack of knowledge on the performance of the sensor.

Furthermore, a lack of measurements is in itself an epistemic uncertainty. When this uncertainty is aleatory, it is probably the easiest to understand and model, if enough data are available on the exact repeatability of measurements.

- **Geometric** uncertainty is a form of aleatory uncertainty on the exact repeatability of the manufacturing of parts and systems.
- **Parameter** uncertainty can be either aleatory or epistemic and refers to the variability of model parameters and boundary conditions.
- **Numerical** (or algorithmic) uncertainty, also known as numerical errors, refers to different types of uncertainty related to each particular numerical scheme, and to the machine precision (including clock drifts).
- **Human** uncertainty is difficult to capture as it has both aleatory and epistemic elements and is dependent on our conscious and unconscious decisions and reactions. It includes the possible variability of goals and requirements due to human decisions.

Uncertainty can be associated to the purpose for which the system is created (uncertainty in the requirements and objectives), in its operations and functionalities, in the phase of its development or in its temporal evolution. Finally, it is important to distinguish between uncertainty affecting functional and physical constraints, and thus defining the feasibility or reliability of a system, and uncertainty affecting the cost or objective function that quantifies the performance of a system. This latter uncertainty defines the robustness of a system. In the following we will consider more specifically epistemic uncertainty in cost function and constraints.

17.1.2 From Design by Analysis to Robust Design Optimisation

In the classical approach to engineering design, *Design by Formula*, the active work of engineers was required throughout the whole design process. In the more recent *Design by Analysis* [26] approach, the development of software analysis tools (numerical methods) shortened the design process and enabled a better understanding of the problem without the use of expensive experimental analyses. The design and associated decision-making process were still performed by engineers, but the analysis of different configurations was automatised by numerical procedures. A further advancement was introduced with *Design by Optimisation* [27], where numerical optimisation tools were coupled with numerical simulations to automatically identify globally, or locally, optimal design solutions. Finally, in the last two decades an increasing attention has been devoted to tackle optimisation under uncertainty. *Design for Reliability and Robustness* [10, 11, 28–31] is radically changing systems engineering, making designers and decision makers able to handle higher degrees of complexity.

Design for robustness means to look for a solution for which the value of the objective function/s (or performance index/es) is optimal under uncertainty while design for reliability means to look for a solution that increases the probability of

satisfying the constraints under uncertainty. In both cases, from a computational point of view, uncertainty can be treated in two different ways

1. *deterministically*, where the quantity of interest is computed for a deterministic variation of, for example, the uncertain parameters within some deterministically defined sets;
2. *non-deterministically*, where some uncertainty measures are computed as a function of the uncertain parameters;

Consider now a generic deterministic optimisation problem associated to the design of a system characterised by the performance index f and decision vector \mathbf{d} :

$$\min_{\mathbf{d} \in D} f(\mathbf{d}) \quad (17.1)$$

subjected to

$$g_j(\mathbf{d}) \leq 0 \quad j = 1, \dots, r \quad (17.2)$$

where D is a design space and g_j are constraint functions. If the system is affected by some form of uncertainty, characterised by the uncertainty vector $\mathbf{u} \in U$, with U the uncertainty space, the problem can be re-formulated as:

$$\min_{\mathbf{d} \in D} \phi(\mathbf{d}, \mathbf{u}) \quad (17.3)$$

subjected to

$$\gamma_j(\mathbf{d}, \mathbf{u}) \leq 0 \quad j = 1, \dots, r \quad (17.4)$$

where ϕ and γ_j are some measures that account for the effect of \mathbf{u} , respectively, on the quantities of interest f and g_j . If uncertainty is treated deterministically and there are no constraints, then ϕ in problem (17.3) can be written as:

$$\phi(\mathbf{d}, \mathbf{u}^*) = \sup_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \quad (17.5)$$

where \mathbf{u}^* is the uncertain vector for which f attains the maximum value. This approach is called *robust regularisation* in [32]. In [32] the function ϕ is defined as the worst case scenario in a neighbourhood $U(\epsilon)$ where ϵ is called the *regularisation parameter*. This approach can also be found in [33–36] and can be generalised, as in Eq. (17.5), to include the whole uncertainty space. In this case the value ϕ is the global worst case scenario.

If there is enough information to model uncertainty with probability distributions, a probabilistic approach can be used. Two examples of probabilistic measures of robustness are:

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

$$F_\phi(\mathbf{d}) = E[\phi(f)|\mathbf{d}] \quad (17.6)$$

where the value of the expectation is conditional to the choice of the design (or decision) vector \mathbf{d} . Different definitions of the function ϕ have been proposed in the literature, see [37] and [38] for some examples. When the utility function is simply f one can account for both the expected value of f and its variance σ with the two-objective problem:

$$\begin{aligned} & \min_{\mathbf{d}} E(f|\mathbf{d}) \\ & \min_{\mathbf{d}} \sigma(f|\mathbf{d}) \end{aligned} \quad (17.7)$$

Methods for solving problem (17.7) can be found in [38–42].

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

$$\max_{\mathbf{d}} Pr(f < q|\mathbf{d}). \quad (17.8)$$

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

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

When there is not enough information to quantify uncertainty with a probability distribution, alternative theories have been used to derive a proper quantification. In this chapter we will focus on the use of Evidence Theory (or DST) already introduced by previous authors in the context of engineering applications [43].

However, it is here worthwhile to briefly mention another approach using Fuzzy Set Theory. Fuzzy Set Theory describes the feasibility of a solution by the membership function μ_c (in the interval [0 1]) instead of using the extremes values 0 or 1 as with the crisp sets. Simoes [44] and Campos et al. [45] propose to use μ_c and an other membership function μ_f for the objective function f , considering $\mu_f = 1$ for the unconstrained optimum and $\mu_f = 0$ for the worst case. Finally the robust solution can be found maximising the minimum of the two defined membership functions:

$$\max(\min[\mu_c, \mu_f]) \quad (17.9)$$

17.2 Evidence Theory

Evidence Theory, known also as Dempster–Shafer Theory (DST) and as Theory of Belief functions, was developed by Shafer (1976) in [19] starting from Dempsters

original work [18]. DST belongs to the class of Imprecise Probabilities Theories that generalise the classical concept of probability by considering the different nature and manifestations of uncertainty. DST allows one to treat both aleatory and epistemic uncertainty with partial information and lack of knowledge. A discussion on the main advantages and disadvantages of DST can be found in [46] and will not be presented in this chapter. On the contrary, this chapter is only concerned with the computation of robust and reliable solutions once uncertainty is quantified with DST.

The central idea of DST can be understood with the following example. Imagine that a spacecraft S , initially in orbit around the Earth, falls down in a not precise zone of Europe. The defence commission recruits a group of m experts, $E^m = \{E_1, E_2, \dots, E_m\}$, to predict where the spacecraft will fall. Imagine that a subset of them, $E^n \subset E^m$, believes that S will fall in some crash area $L_i \quad \forall i \mid E_i \in E^n$ and the others E^{m-n} believe that the S will fall in the ocean $L_j = \emptyset \quad \forall j \mid E_j \in E^{m-n}$. The question posed by the commission is: “What is evidence in support that S will fall in the specific land-area A ?” Each expert, considering all the available information and their knowledge, gives one of these three possible answers: (i) the whole crash area is included in A ($L_i \subset A$), (ii) S could fall in A ($L_i \cap A \neq \emptyset$) or (iii) is not possible for the S to fall in A ($L_i \cap A = \emptyset$). Imagine that q experts in E_m agree with (ii) and in particular k of them answer (i), where $q \geq k \in E_m$. The commission can then reconstruct the *belief* that S will fall inside A , $\frac{k}{m}$, and the *plausibility*, $\frac{q}{m}$, that S might fall inside A . In this case the commission assumes that all experts are equally credible, i.e. that the evidence they provide has equal weight. If $m - n$ experts do not believe there is a dangerous situation, belief and plausibility became: $Bel = \frac{k}{n}$ and $Pl = \frac{q}{n}$. If some experts are recognised to be more reliable than others, the commission can use a set of weights, $\omega_1, \omega_2, \dots, \omega_n$, to assign more or less belief to the statement of each expert. In this case the cumulative belief and plausibility values become

$$Bel = \frac{1}{K} \sum_{\omega_i \mid L_i \subseteq A} \omega_i \quad (17.10)$$

$$Pl = \frac{1}{K} \sum_{\omega_i \mid L_i \cap A \neq \emptyset} \omega_i \quad (17.11)$$

where K is the normalising factor:

$$K = 1 - \sum_j \omega_j, \quad E_j \cap A \neq \emptyset \quad (17.12)$$

Note that the true probability P of the satellite to fall in A is expected to be $Bel \leq P \leq Pl$. However, no expert has an exact quantification of this probability, hence the commission can only rely on their informed opinions and in the credibility of each expert. In DST the weights are called *basic probability assignment* or *bpa*.

17.2.1 Frame of Discernment, Power Set and Evidence

There are three concepts at the basis of DST: the *Frame of Discernment*, the *Power Set* and the *Evidence*. The *frame of discernment* Θ is the set of all the mutually exclusive and collectively exhaustive elementary events (or hypothesis) $\theta_i, i = 1, \dots, |\Theta|$:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_{|\Theta|}\} \quad (17.13)$$

All the possible events (or hypotheses) could be overlapping or nested, but in the *frame of discernment* only the finest division of them is considered. The frame of discernment Θ could be thought as the finite sample space in general Probability Theory. From the *frame of discernment* one can define the *power set* $2^\Theta = (\Theta, \cup)$ by considering all possible combinations of the elements of Θ :

$$\Omega = 2^\Theta = \{\emptyset, \{\theta_1\}, \dots, \{\theta_{|\Theta|}\}, \{\theta_1, \theta_2\}, \dots, \{\theta_1, \theta_2, \dots, \theta_i\}, \dots, \{\theta_1, \theta_3\}, \Theta\} \quad (17.14)$$

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

The degree of belief, or evidence, is quantified by the *bpa* that assigns a value $m \in [0, 1]$ to each subset of Ω :

$$m : 2^\Theta \rightarrow [0, 1] \quad (17.15)$$

where the function m has to satisfy the following conditions:

$$m(\omega) \geq 0, \forall \omega \in \Omega \quad (17.16)$$

$$m(\omega) = 0, \forall \omega \notin \Omega \quad (17.17)$$

$$m(\emptyset) = 0 \quad (17.18)$$

$$\sum_{A \in 2^\Theta} m(A) = 1 \quad (17.19)$$

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

An interesting feature of DST is the ability to properly model ignorance and to differentiate between pure randomness and lack of knowledge. Furthermore, as in the example presented in the previous section, one can very naturally quantify uncertainty through expert opinions without a precise probability assignment.

Consider for example, that only two events θ_1 and θ_2 are possible. In probability theory the knowledge on the probability of the realisation of one of the two events is sufficient to make a statement on the realisation of the other event. In other

words $m(\theta_1) + m(\theta_2) = 1$. DST, instead, does not make any assumption on the information available for each of the two events and contemplates also the ignorance statement $\theta_1 \cup \theta_2$, or $m(\theta_1) + m(\theta_2) \leq 1$. For this reason, in *DST* the additive rule ($m(\{\theta_1, \theta_2\}) = m(\{\theta_1\}) + m(\{\theta_2\})$) does not hold and is replaced by a super-additive ($m(\{\theta_1, \theta_2\}) > m(\{\theta_1\}) + m(\{\theta_2\})$) or sub-additive ($m(\{\theta_1, \theta_2\}) < m(\{\theta_1\}) + m(\{\theta_2\})$) rule. The immediate consequence is that $m(A) + m(\neg A) \leq 1$, i.e. the probability on a set and on its negation do not sum to one.

17.2.2 Belief and Plausibility

At the beginning of this section we already informally introduced the idea of belief *Bel* and plausibility *Pl* with a simple example. Now that *frame of discernment*, *power set* and *bpa* are defined, one can more formally calculate the belief and plausibility associated to a particular quantity of interest $f : U \rightarrow \mathcal{R}$, where U is the uncertain space and is here considered to be equal to the *power set*: $U = 2^\Theta$. Calling Φ a target region for the quantity of interest f , one is interested in measuring the degree of belief in the realisation of $f(x) \in \Phi$. If one defines the preimage:

$$A = \{x \in U \mid f(x) \in \Phi\}, \tag{17.20}$$

the total degree of belief associated to (17.20) can be computed by collecting all the pieces of evidence that either fully or partially support that statement. Given the set A defined in (17.20) and the focal elements ω , the cumulative functions *Bel* and *Pl* are defined as:

$$Bel(A) = \sum_{\omega_i \subseteq A} m(\omega_i) \tag{17.21}$$

$$Pl(A) = \sum_{\omega_i \cap A \neq \emptyset} m(\omega_i) \tag{17.22}$$

where ω_i is the generic *FE* inside $\langle F, m \rangle$ and the plausibility measure *Pl* is the dual function of the belief measure *Bel*:

$$Pl(A) = 1 - Bel(\bar{A}) \tag{17.23}$$

with \bar{A} is the complement to A . From Eq. (17.21) one can see that the *Bel* function is the sum of all the pieces of evidence that completely support the statement $f \in \Phi$, whereas the *Pl* function is the sum of all the pieces of evidence that partially support the statement $f \in \Phi$: this means that $m(\omega_i)$ is added to *Bel* only if all possible realisations of $x \in \omega_i$ belong to A , on the contrary $m(\omega_i)$ is added to *Pl* if at least one realisation of $x \in \omega_i$ belongs to A . This suggests that the *Evidential Interval (EI)* [*Bel Pl*] is a measure of the degree of ignorance on the probability

of a realisation of x . When enough information is available, the distribution of x is known and one can exactly quantify if a realisation of x belongs to A or not. In this case all FEs collapse to singletons and the following condition holds true:

$$Bel = Pl = P \tag{17.24}$$

The functions Bel and Pl are monotonic non-additive measures and have following properties:

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

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

and

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

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

Due to conditions 3, Belief and Plausibility are also called *monotone capacities of order ∞* .

17.3 Robust Optimisation with Evidence Theory

The central idea underneath Evidence-Based Robust Optimisation is to maximise the belief in statement (17.20). This condition alone, however, is not enough to qualify the realisations of f . In fact, imagine that f is a performance indicator, then the condition $f \in \Phi$ alone would not say much on the optimality of the values of f .

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

$$\begin{aligned} & \min_{\nu \in \mathbb{R}} \nu \\ & s.t. \\ & f(\mathbf{d}, \mathbf{u}) \leq \nu \end{aligned} \tag{17.25}$$

that leads to the robust optimisation problem [47]:

$$\begin{aligned} & \max_{\mathbf{d} \in D} \text{Bel}(f(\mathbf{d}, \mathbf{u}) \leq \nu) \\ & \min_{\nu \in \mathbb{R}} \nu \end{aligned} \quad (17.26)$$

Problem (17.26) can be extended to include constraints in three different forms:

$$\begin{aligned} & \max_{\mathbf{d} \in D} \text{Bel}(f(\mathbf{d}, \mathbf{u}) \leq \nu) \\ & \min_{\nu \in \mathbb{R}} \nu \\ & \text{s.t.} \\ & C(\mathbf{d}, \mathbf{u}) \leq \nu_c \end{aligned} \quad (17.27)$$

$$\begin{aligned} & \max_{\mathbf{d} \in D} \text{Bel}(f(\mathbf{d}, \mathbf{u}) \leq \nu) \\ & \min_{\nu \in \mathbb{R}} \nu \\ & \text{s.t.} \\ & \text{Bel}(C(\mathbf{d}, \mathbf{u}) \leq \nu_c) > 1 - \epsilon \end{aligned} \quad (17.28)$$

$$\begin{aligned} & \max_{\mathbf{d} \in D} \text{Bel}(f(\mathbf{d}, \mathbf{u}) \leq \nu_f) \\ & \max_{\mathbf{d} \in D} \text{Bel}(C(\mathbf{d}, \mathbf{u}) \leq \nu_c) \\ & \min_{\nu_f \in \mathbb{R}} \nu_f \\ & \min_{\nu_c \in \mathbb{R}^{n_c}} \nu_c \end{aligned} \quad (17.29)$$

Problem (17.27) introduces the deterministic constraint vector function C , problem (17.28) introduces a set of constraints on the belief that the constraints are satisfied, while problem (17.29) tries to maximise the belief that the constraints are satisfied. Note that problem (17.28) might not have any solution even if problem (17.27) has a solution because the constraint on the belief of the satisfaction of the constraints implies that constraints need to be satisfied for a set of values and not for a single one. Because of condition (17.23) it is clear that one can derive an equivalent formulation with Pl .

Although the formulation of an evidence-based robust optimisation problems looks simple, the solution even of the unconstrained problem (17.26) is far from trivial. In fact the computation of Bel presents two major difficulties:

1. In order for a focal element ω to be included in the calculation of the belief, the following condition must be true:

$$\max_{\mathbf{u} \in \omega} f \leq \nu \quad (17.30)$$

which implies solving a number of (global) maximisation problems equal to the number of focal elements.

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

In [48] the authors present three approaches to solve problem (17.26):

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

$$I_i = |\{j | Bel(\mathbf{d}_j) > Bel(\mathbf{d}_i) \wedge \nu_j < \nu_i, j = 1, \dots, n_{pop} \wedge j \neq i\}| \quad (17.31)$$

with $|\cdot|$ the cardinality and n_{pop} the number of design vectors. But this approach has two main problems: each design vector in (17.31) is related to a Belief— ν curve, and different design vectors could give the same Pareto front.

- The *step method* reduces the computational effort solving a single objective problem: an initial \mathbf{d} is chosen that corresponds to a threshold ν_1 with $Bel = 1$ and then the threshold is reduced step by step running, for each new ν_k , a local optimisation and maximising the corresponding Belief. The new optimisation is started from the previous optimal \mathbf{d} configuration and a local optimiser is used; this reduces the possibility to evaluate the real global optimum, but it is on the other hand a necessary simplification to avoid the explosion in computational time.
- The *cluster approximation*, finally, looks at the whole search space (design and uncertainty) and for different thresholds ν_i clusters all the possible sets, in $\mathbf{D} \times \mathbf{U}$, that satisfy the condition: $f < \nu_i$. For each ν_i and design, then, the belief can be easily evaluated adding the FEs included in the cluster and finally the \mathbf{d} that maximise the belief approximation is chosen.

One can circumvent the difficulties with the calculation of Bel by taking a particular value of ν such that $\Omega = U \implies Bel(\Omega) = 1$. From the point of view of the performance index f , this is the worst case scenario because it corresponds to the situation in which $f \leq \nu \forall \mathbf{u} \in U$, in other words ν is a global maximum for f .

Problem (17.26) then translates in the classical min-max (robust optimisation) problem:

$$\min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \quad (17.32)$$

or

$$\begin{aligned} &\min_{\mathbf{d} \in D} f(\mathbf{d}) \\ &s.t. \\ &f(\mathbf{d}) = \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \end{aligned} \quad (17.33)$$

If $\Omega \neq U$, one can still write the constrained min-max problem:

$$\begin{aligned}
& \min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \\
& \text{s.t.} \\
& C(\mathbf{d}, \mathbf{u}) \leq 0
\end{aligned} \tag{17.34}$$

where this time $\Omega = \{\mathbf{u} | C(\mathbf{d}, \mathbf{u}) \leq 0\}$ and $C : U \times D \rightarrow \mathbb{R}^{n_c}$ is a vector function. In this case, the value of Bel is not 1 and the Bel associated to the solution of problem (17.34) needs to be evaluated a posteriori. If one requires the problem to be globally reliable and not just globally robust, then the constrained min-max problem becomes

$$\begin{aligned}
& \min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \\
& \text{s.t.} \\
& \max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) \leq 0
\end{aligned} \tag{17.35}$$

with $\Omega = \{\mathbf{u} | \max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) \leq 0\}$. Note that problem (17.34) is not equivalent to problem (17.26). In fact, consider the constrained problem:

$$\begin{aligned}
& \min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \\
& \text{s.t.} \\
& f(\mathbf{d}, \mathbf{u}) - \nu \leq 0
\end{aligned} \tag{17.36}$$

In this case the optimal solution does not necessarily correspond to a maximum Bel value because the Bel value is the sum of all focal elements that satisfy $f \leq \nu$ while the minimisation with respect to \mathbf{d} would only consider a single realisation of \mathbf{u} .

Problems (17.34) can be readily extended to the case in which $\mathbf{f} = [f_1, f_2, \dots, f_m]^T$ is a vector function, as follows:

$$\begin{aligned}
& \min_{\mathbf{d} \in D} [\max_{\mathbf{u} \in U} f_1, \max_{\mathbf{u} \in U} f_2, \dots, \max_{\mathbf{u} \in U} f_m]^T \\
& \text{s.t.} \\
& \max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) \leq 0
\end{aligned} \tag{17.37}$$

Likewise one can extend problem (17.26) to the multi-Belief case:

$$\begin{aligned}
& \max_{\mathbf{d} \in D} Bel(f_1 \leq \nu_1) \\
& \max_{\mathbf{d} \in D} Bel(f_2 \leq \nu_2) \\
& \dots \\
& \max_{\mathbf{d} \in D} Bel(f_m \leq \nu_m) \\
& \min_{\nu_1 \in \mathbb{R}} \nu_1 \\
& \min_{\nu_2 \in \mathbb{R}} \nu_2 \\
& \dots \\
& \min_{\nu_m \in \mathbb{R}} \nu_m
\end{aligned} \tag{17.38}$$

Note that the dual to problem (17.37) is the minimisation problem:

$$\begin{aligned}
& \min_{\mathbf{d} \in D} [\min_{\mathbf{u} \in U} f_1, \min_{\mathbf{u} \in U} f_2, \dots, \min_{\mathbf{u} \in U} f_m]^T \\
& \text{s.t.} \\
& \max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) \leq 0
\end{aligned} \tag{17.39}$$

In both problem (17.37) and (17.39) the assumption is that the realisations of \mathbf{u} that maximise or minimise the components of \mathbf{f} are independent.

17.3.1 Optimising the Worst Case Scenario

In the literature, a number of optimisation techniques have been proposed to solve problem (17.32). Some are based on mathematical programming [49–60] while others are using heuristic methods [61–64]. In particular, interesting results have been obtained with evolutionary algorithms (EAs). As in [65], three main categories can be identified:

- algorithms for discrete min-max problems evaluating all [66] or a subset [67, 68] of the uncertain scenarios;
- algorithms directly solving the nested problem $\max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u})$ within Eq.(17.33) [69, 70]. This approach is highly expensive for costly objective functions;
- the last approach defined in [65] is the co-evolution strategy: it emulates the dynamic evolution driven by natural selection where each organism has to continuously adapt to the other: two different populations are developed in parallel while information is shared between them. Some ideas are developed in [71–73] and [74] extends the previous three papers for problems that are not constrained to satisfy the symmetrical condition $\min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) = \max_{\mathbf{u} \in U} \min_{\mathbf{d} \in D} f(\mathbf{d}, \mathbf{u})$.

Furthermore some hybrid methods exist that mix evolutionary strategies and mathematical programming. Given the considerable computational cost required to solve problem (17.32) some authors proposed methods to reduce the number of function evaluations through the use of surrogate models: [65, 75, 76] are Surrogate-Assisted Evolutionary Algorithms (SAEA) and [77] proposes a kriging surrogate.

The algorithm proposed in this section solves the more general constrained problem (17.35) using a combination of optimisation and restoration loops similar to what can be found in [77]. An extensive description of the approach and its performance can be found in [78].

The optimisation loop minimises $\max_{\mathbf{u}_e \in A} f(\mathbf{d}, \mathbf{u}_e)$ over the decision vector \mathbf{d} and the restoration loop maximises $f(\mathbf{d}^*, \mathbf{u})$ over the uncertainty vector \mathbf{u} for $\mathbf{d}^* = \arg \min_{\mathbf{d}} \max_{\mathbf{u}_e \in A} f(\mathbf{d}, \mathbf{u}_e)$. The vectors \mathbf{u}_e are taken from the space of the maxima over U . The constraint vector C defines the admissible set for both \mathbf{d} and \mathbf{u} . In particular, in order for the solution to be robust against all realisations of \mathbf{u} one has

to ensure that the maximum value of C over U is admissible. The optimisation and restoration loops then become as follows:

1. [Optimisation] Given a set of maxima $A = A_u \cup A_c$, solve the constrained minimisation problem:

$$\begin{aligned} \min_{\mathbf{d} \in D} \max_{\mathbf{u} \in A} f(\mathbf{d}, \mathbf{u}) \\ \text{s.t.} \\ \max_{\mathbf{u} \in A} C(\mathbf{d}, \mathbf{u}) \leq 0 \end{aligned} \quad (17.40)$$

where A_u is the set of maxima of f and A_c is the set of maxima of C .

2. [Restoration] Given the solution of problem (17.40), \mathbf{d}^* , solve the two maximisation problems:

$$\begin{aligned} \max_{\mathbf{u} \in U} f(\mathbf{d}^*, \mathbf{u}) \\ \text{s.t.} \\ C(\mathbf{d}, \mathbf{u}) \leq 0 \end{aligned} \quad (17.41)$$

$$\max_{\mathbf{u} \in U} C(\mathbf{d}^*, \mathbf{u}) \quad (17.42)$$

The solution of problem (17.41), $\mathbf{u}_{a,f}$, is added to the archive A_u and the solution of problem (17.42), $\mathbf{u}_{a,C}$ is added to A_c if $\max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) > 0$. Note that problem (17.42) has to be understood as a maximisation for every constraint function in C and not as a vector optimisation. This approach pushes the optimiser to find design solutions that are feasible for all values of the uncertain variables. If a feasible solution cannot be found, the constraints are relaxed (line 15), by defining the new constraint $C^* = C + \epsilon$ with ϵ the minimum constraint violation over U .

The optimisation and restoration loops are repeated one after the other for a prescribed number of iterations and all \mathbf{d}^* and associated maxima A are stored in a global archive A_g . The global archive is then used to perform a cross-check of the solutions. Given a finite number of iterations, one might obtain a solution \mathbf{d}^* associated to non-globally optimal value of $\mathbf{u}_e \in A$. In order to mitigate this problem one can evaluate f and C taking multiple pairs $\mathbf{d}^*, \mathbf{u}_e$ taken from the archive A . The overall procedure is summarised in Algorithm 1. First the design vector $\bar{\mathbf{d}}$ and the archives— A_u, A_c, A_d —are initialised and a constrained maximisation over the uncertain domain U is run kipping fixed $\bar{\mathbf{d}}$ (lines 1 and 2). Then the inner and outer loops are alternated until the maximum number of iteration is reached (lines 3–22). In particular, the archive of the design vectors \mathbf{d} , A_d , is updated after each outer loop (line 6) while the archives of the uncertainty vectors \mathbf{u}_F —from the maximisation of the objective function—and \mathbf{u}_C —from the maximisation of the constraint violation—are updated after each inner loop (respectively, in lines 10 and 13) if they are not already saved in the archives. During the last loops of the

algorithm the relaxation procedure could be activated if the condition expressed in line 13 is satisfied: a fixed number of iterations—arbitrarily lower than the maximum allowed—has to be reached and none of the solution saved in the whole archive $A = A_F \cup A_C$ has to be feasible in all the uncertainty domain U . If this happens, a small violation ϵ of the constraint C is accepted and increased as long as a feasible solution is obtained. The relaxation procedure is helped by the elimination from the archive A_C of all the vectors \mathbf{u} previously saved with to a constraint violation smaller than the actual ϵ (line 17).

Algorithm 1 Constrained minmax

```

1: Initialise  $\bar{\mathbf{d}}$  at random and run  $\mathbf{u}_a = \operatorname{argmax} F(\bar{\mathbf{d}}, \mathbf{u})$  s.t.  $C(\mathbf{d}_{min}, \mathbf{u}) \leq 0$ 
2:  $A_u = A_u \cup \{\mathbf{u}_a\}$ ;  $A_c = \emptyset$ ;  $A_d = \emptyset$ 
3: while  $N_{fval} < N_{fval}^{max}$  do
4:   Outer loop:
5:    $\mathbf{d}_{min} = \operatorname{argmin}_{\mathbf{d} \in D} \{\max_{\mathbf{u} \in A_u \cup A_c} F(\mathbf{d}, \mathbf{u})\}$  s.t.
      $\max_{\mathbf{u} \in A_u \cup A_c} C(\mathbf{d}, \mathbf{u}) \leq 0$ 
6:    $A_d = A_d \cup \{\mathbf{d}_{min}\}$ 
7:   Inner loop:
8:    $\mathbf{u}_{a,F} = \operatorname{argmax}_{\mathbf{u} \in U} F(\mathbf{d}_{min}, \mathbf{u})$  s.t.  $C(\mathbf{d}_{min}, \mathbf{u}) \leq 0$ 
9:    $\mathbf{u}_{a,C} = \operatorname{argmax}_{\mathbf{u} \in U} C(\mathbf{d}_{min}, \mathbf{u})$ 
10:   $A_u = A_u \cup \{\mathbf{u}_{a,F}\}$ 
11:  if  $N_{fval} < N_{fval}^{relaxation} \vee \exists \mathbf{d} \in A_d$  t.c.  $\max_{\mathbf{u} \in U} C(\mathbf{d}, \mathbf{u}) \leq 0$  then
12:    if  $\max_{\mathbf{u} \in U} C(\mathbf{d}_{min}, \mathbf{u}) > 0$  then
13:       $A_c = A_c \cup \{\mathbf{u}_{a,C}\}$ 
14:    end if
15:  else
16:    update  $\epsilon$ 
17:     $A_c = \{A_c \setminus \mathbf{u}_{a,C} \mid C(\mathbf{d}_{min}, \mathbf{u}) \leq \epsilon\}$ 
18:    if  $\max_{\mathbf{u} \in U} C(\mathbf{d}_{min}, \mathbf{u}) > \epsilon$  then
19:       $A_c = A_c \cup \{\mathbf{u}_{a,C}\}$ 
20:    end if
21:  end if
22: end while

```

17.4 Belief Curve Reconstruction

The solution of the worst case scenario provides some key information on the system under investigation. It defines the range of variability of ν for problem (17.26), it provides a quantification of the extreme case and a possible countermeasure and explores the space of the maxima over U which represents the set of extreme cases for f and C . However, the most robust solution is also the most conservative and does not account for the belief associated to each event. The solution of (17.26) requires, instead, an efficient calculation of Bel for events different from the worst case. In this section we consider a few approaches to calculate an approximated value of Bel for a range of values of ν . We can group these approaches in

three categories: sampling methods, dimensionality reduction methods and outer approximation methods.

17.4.1 Belief Estimation by Sampling

In the literature one can find a handful of sampling-based methods for the computation and estimation of *Bel* [22, 79–81]. As a representative example, in [22], the authors suggest the use of the density function:

$$d_j(u_j) = \sum_{k=1}^{n(j)} \delta(u_j|I_{jk}) bpa_{jk}(I_{jk}) / (b_{jk} - a_{jk}) \quad (17.43)$$

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

$$\delta(x_j|I_{jk}) = \begin{cases} 1 & \text{if } x_j \in I_{jk} \\ 0 & \text{otherwise;} \end{cases} \quad (17.44)$$

then the sampling distribution is

$$d(\mathbf{x}) = \prod_{j=1}^{nU} d_j(u_j) \quad (17.45)$$

Distribution (17.45) explores adequately the uncertain space giving more importance to the focal elements with higher *bpa* and then sampling uniformly inside them. Samples can be generated with a Latin Hypercube scheme and propagated through the system model in order to build a response surface, for example, through a non-parametric regression model, that can either directly approximate the *Bel* or the quantity of interest from which one can calculate the *Bel*.

The main problem with sampling is that a correct calculation of the *Bel* requires the maximisation of the quantity of interest within each focal element. If sampling is used to directly estimate the *Bel* one obtains a potentially efficient approach but an approximation that can be significantly poor. Furthermore, this type of approximation provides estimated *Bel* values that are always better than the actual ones, leading to an overconfidence in the realisation of an event. A possible mitigation of this problem was recently proposed in [82] to address the solution of optimal control problem under epistemic uncertainty. In [82] the value of the *Bel* was approximated with the surrogate of a weighted integral obtained by sampling the space of the focal elements. The integral was elevated to an exponent factor k , the higher k the more the integral was resembling the actual *Bel*. Furthermore,

the surrogate was periodically updated to identify the threshold values where the approximation was the closest to the true *Bel*. These two improvements allow one to insert selected values of the surrogate in the optimisation loop and improve the *Bel* at a discrete number of thresholds ν .

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

17.4.2 Dimensionality Reduction

Another class of approaches tries to reduce the number of focal elements upfront. The general idea is to define criteria to sort the *FEs* by their importance and then approximate the m -function in Eq. (17.15) to a m' -function with a lower number of *FEs*. A few examples follow:

- The Bayesian approximation proposed by Voorbraak's [83] produces a discrete probability distribution: the new mass function m' considers only the singleton subsets θ_i in the *Power Set* 2^Θ .
- Dubois and Prade's consonant approximations [84]
- The k - l - x method proposed by Tessem [85] uses the *bpa*'s as sorting criterion. The approximating $m'=m_{klx}$ then includes only the p *FEs* with higher *bpa*, where $k < p < l$, and such that the sum of the masses of the deleted *FEs* is less than x . A normalisation method is finally used to redistribute the total mass of the deleted *FEs* to the remaining ones.
- The *summarisation method* takes the first p focal elements with the highest *bpa*, as in [85], and lumps together all the remaining ones in a single focal element with a *bpa* that is the sum of their *bpa*'s.
- The *D1 method* [86] beside the criteria of *mass* introduces also the *cardinality*.
- The *Batch approximation method* and *Iterative approximation method* [87] suggest, instead, that *mass* and *cardinality* are not sufficient to discriminate which *FE* to take and which one to discard. Then in the paper a *non-redundancy* measure is presented based on the definition of distance between two *FEs* as proposed by Dencœux in [88].
- for some heuristic methods refer to [89].

17.4.3 Outer Belief Estimation via Evolutionary Binary Tree

It is generally desirable to have a method that produces estimated values of *Bel* that are lower than the actual one. We call these estimated values: *outer approximations*.

Furthermore, it is desirable to have a method that iteratively converges to the exact value with a sequence of outer approximations.

In [20], *Bel* and *Pl* functions were approximated, for any threshold $\nu \in [\nu_{min}, \nu_{max}]$, with an Evolutionary Binary Tree (*EBT*) algorithm. *EBT* consists in an iterative bisection of the uncertainty space \bar{U} , where \bar{U} is the collection of all the focal elements into a unit hypercube. The assumption is that focal elements are formed from a number of disjoint or overlapping intervals along every dimension. The mapping between the uncertainty space U and the unit hypercube \bar{U} is given by an affine transformation.

Every time a subset $B_{i,j}$ of \bar{U} is bisected (where i represents the level of the tree and j the index of the branch at level i) the maximum of f in one of two new halves is computed. The other half inherits the maximum of f in $B_{i,j}$. At every iteration, subsets are pruned from the tree if the maxima belonging to those subset are below the threshold ν and the cumulative *bpa* associated to those subsets is added to the calculation of *Bel*. In the original formulation in [20] a subset was bisected along the longest edge and along the extreme of the interval that was closest to the middle point of the edge. Thus the partitioning rule was agnostic of the distribution of focal elements.

The partition of the unit hypercube eventually reduces to all the focal elements whose maxima are above the threshold ν . Hence this method eventually converges to the true *Bel* and produces better and better outer approximations of its value as the partitioning process progresses.

In [21] a more sophisticated and efficient partitioning scheme was proposed that exploits the distribution of the maxima and the *bpa* associated to each focal element. The method is based on the variance of the distribution of the maxima in the subsets $B_{i,j}$ generated by the partitioning of \bar{U} . The partitioning rule bisects the subset along the edge (or dimension) where distribution of the maxima displays the highest variance. The process is finally stopped when a maximum number of evaluations of f is reached or when the difference between the value of *Bel* after two subsequent iterations is below a given tolerance.

17.4.4 Outer Belief Estimation via Decomposition

In some cases the structure and nature of the function f can be exploited to drastically reduce the computation of $Bel(f \leq \nu)$ in (17.26). In particular, in this section we consider the model introduced in [90] and the decomposition method proposed in [90–92].

17.4.4.1 Evidence Network Models

Consider a generic complex system with N sub-systems connected with a known topology. We model this complex system as a non-directed graph or network where

node i is characterised by the value function g_i and exchanges information with node j via the exchange function h_{ij} . Following the definition of the design vector \mathbf{d} and uncertain vector \mathbf{u} introduced in the previous sections, one can define the total value of the network as:

$$f(\mathbf{d}, \mathbf{u}) = \sum_{i=1}^N g_i(\mathbf{d}, \mathbf{u}_i, \mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})) \tag{17.46}$$

where $\mathbf{h}_i(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})$ is the vector of scalar functions $h_{ij}(\mathbf{d}, \mathbf{u}_i, \mathbf{u}_{ij})$, $j \in J_i$ and J_i is the set of indexes of nodes connected to the i -th node; \mathbf{u}_i are the uncertain variables of sub-system i not affecting any other sub-systems and \mathbf{u}_{ij} are the uncertain variables affecting both sub-systems i and j . Please note that accordingly to our notation $\mathbf{u}_{ij} = \mathbf{u}_{ji}$.

As an example, a fully connected network with 3 nodes can be represented as in Fig. 17.1, the value function f in this case is

$$\begin{aligned} f(\mathbf{d}, \mathbf{u}) = & g_1(\mathbf{d}, \mathbf{u}_1, h_{12}(\mathbf{d}, \mathbf{u}_1, \mathbf{u}_{12}), h_{13}(\mathbf{d}, \mathbf{u}_1, \mathbf{u}_{13})) \\ & + g_2(\mathbf{d}, \mathbf{u}_2, h_{21}(\mathbf{d}, \mathbf{u}_2, \mathbf{u}_{12}), h_{23}(\mathbf{d}, \mathbf{u}_2, \mathbf{u}_{23})) \\ & + g_3(\mathbf{d}, \mathbf{u}_3, h_{31}(\mathbf{d}, \mathbf{u}_3, \mathbf{u}_{13}), h_{32}(\mathbf{d}, \mathbf{u}_3, \mathbf{u}_{23})) \end{aligned} \tag{17.47}$$

We then call \mathbf{u}_i *uncoupled* variables because they influence only sub-system i and \mathbf{u}_{ij} *coupled* variables because they influence sub-systems i and j ; if the same parameters are shared between nodes i , j and k is $\mathbf{u}_{ij} = \mathbf{u}_{ik} = \mathbf{u}_{jk}$.

We now consider the pair (f, Ω) where $f : \Omega \rightarrow \mathcal{R}$. Furthermore we introduce the two sets Ω_x and Ω_y such that $\Omega = \Omega_x \times \Omega_y$. Consider now two partitions D_x and D_y , respectively, of Ω_x and Ω_y . Given $\delta\Omega_x^p \in D_x$ and $\delta\Omega_y^q \in D_y$ we compute

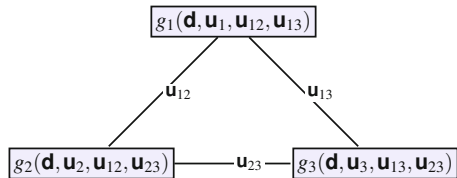
$$\mathbf{y}_0 = \arg \max_{\delta\Omega_y^q} f(\mathbf{x}_0, \mathbf{y}) \tag{17.48}$$

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

$$\mathbf{x}_k = \arg \max_{\delta\Omega_x^p} f(\mathbf{x}, \mathbf{y}_{k-1}) \tag{17.49}$$

$$\mathbf{y}_k = \arg \max_{\delta\Omega_y^q} f(\mathbf{x}_k, \mathbf{y}) \tag{17.50}$$

Fig. 17.1 Evidence network model of a generic system F composed of three sub-systems with coupled variables u_{12} , u_{13} and u_{23}



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

$$(\mathbf{x}_k, \mathbf{y}_k) = \arg \max_{\delta \Omega^{pq}} f(\mathbf{x}, \mathbf{y}) \quad (17.51)$$

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

The ENMs has been constructed under the following properties:

1. The contribution of the coupled variable \mathbf{u}_{ij} to the value f manifests through the scalar functions h_{ij} and h_{ji} .
2. The pair (f, U) is M -decomposable. In particular in this chapter we consider the case in which $M = 0$.

17.4.4.2 Decomposition Method

The decomposition algorithm aims at decoupling the sub-systems over the uncertain variables in order to optimise only over a small subset of the Focal Elements (Algorithm 2); this procedure requires the following steps:

1. Solution of the optimal worst case scenario problem:

$$\min_{\mathbf{d} \in D} \max_{\mathbf{u} \in U} F(\mathbf{d}, \mathbf{u}) \quad (17.52)$$

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

Point 1 has been already discussed in Sect. 17.3.1. In the following the solution of problem (17.52) is represented by the values $\widetilde{\mathbf{d}}$ and $\underline{\mathbf{u}}$ and it is assumed that $\widetilde{\mathbf{d}}$ is already available.

For each coupled vector \mathbf{u}_{ij} a maximisation is run over each Focal Element $\theta_{k,ij} \subseteq \Theta_{ij} \subseteq U$, given $\widetilde{\mathbf{d}}$ and keeping fixed all the other components to $\underline{\mathbf{u}}_{\mathbf{k}}$ and $\underline{\mathbf{u}}_{\mathbf{lm}} \forall k, l, m \setminus \{l, m\} = \{i, j\}$. Taking again the example in Fig. 17.1 we have

$$\begin{aligned} \hat{\mathbf{u}}_{k,12} &= \arg \max_{\mathbf{u}_{12} \in \theta_{k,12}} F(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3, \mathbf{u}_{12}, \underline{\mathbf{u}}_{13}, \underline{\mathbf{u}}_{23}), \forall \theta_{k,12} \subset \Theta_{12} \\ \hat{\mathbf{u}}_{k,13} &= \arg \max_{\mathbf{u}_{13} \in \theta_{k,13}} F(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3, \underline{\mathbf{u}}_{12}, \mathbf{u}_{13}, \underline{\mathbf{u}}_{23}), \forall \theta_{k,13} \subset \Theta_{13} \\ \hat{\mathbf{u}}_{k,23} &= \arg \max_{\mathbf{u}_{23} \in \theta_{k,23}} F(\widetilde{\mathbf{d}}, \underline{\mathbf{u}}_1, \underline{\mathbf{u}}_2, \underline{\mathbf{u}}_3, \underline{\mathbf{u}}_{12}, \underline{\mathbf{u}}_{13}, \mathbf{u}_{23}), \forall \theta_{k,23} \subset \Theta_{23} \end{aligned} \quad (17.53)$$

For easiness in the notation we will indicate with

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

Algorithm 2 Decomposition

```

1: Initialise
2: Uncoupled vectors  $\mathbf{u}_u = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_i, \dots, \mathbf{u}_{m_u}]$ 
3: Coupled vectors  $\mathbf{u}_c = [\mathbf{u}_{12}, \mathbf{u}_{13}, \dots, \mathbf{u}_{ij}, \dots, \mathbf{u}_{m_c}]$ 
4: for a given design  $\tilde{\mathbf{d}}$  do
5:   Compute  $(\tilde{\mathbf{d}}, \underline{\mathbf{u}}_u, \underline{\mathbf{u}}_c) = \arg \max F(\tilde{\mathbf{d}}, \mathbf{u}_u, \mathbf{u}_c)$ 
6:   for all  $\mathbf{u}_{ij} \in \mathbf{u}_c$  do
7:     for all Focal Elements  $\theta_{k,ij} \subseteq \Theta_{ij}$  do
8:        $\hat{F}_{k,ij} = \max_{\mathbf{u}_{ij} \in \theta_{k,ij}} F(\tilde{\mathbf{d}}, \underline{\mathbf{u}}_u, \mathbf{u}_{ij})$ 
9:        $\hat{\mathbf{u}}_{k,ij} = \operatorname{argmax}_{\mathbf{u}_{ij} \in \theta_{k,ij}} F$ 
10:      Evaluate  $bpa(\theta_{k,ij})$ 
11:      Evaluate partial Belief curve  $Bel(F(\mathbf{u}_{ij}) \leq \nu)$ 
12:    end for
13:    for number of samples do
14:      Evaluate  $\Delta Bel^q, \hat{\mathbf{u}}_{k,ij}$  and  $\hat{F}_{k,ij}$ 
15:    end for
16:  end for
17:  for all the combinations of samples do
18:    for all  $\mathbf{u}_i \in \mathbf{u}_u$  do
19:      for all Focal Elements  $\theta_{k,i} \subseteq \Theta_i$  do
20:        Run
21:         $F_{max,k,i} = \max_{\theta_{k,i}} F(\tilde{\mathbf{d}}, \hat{\mathbf{u}}_c, \mathbf{u}_i)$ 
22:        Evaluate  $bpa(\theta_{k,i})$ 
23:      end for
24:    for all the combinations of Focal Elements
25:       $\theta_t \in \Theta_1 \times \Theta_2 \times \dots \times \Theta_{m_u}$  do
26:        Evaluate  $F_{max,k} \leq \nu$ 
27:        Evaluate  $bpa_k$ 
28:      end for
29:    Evaluate the Belief for this sample by constructing collection  $\Gamma_\nu$ 
30:  end for
31: end for

```

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

$$Bel(F(\mathbf{u}_{ij}) < \nu) = \sum_{\theta_{k,ij} | \max_{\mathbf{u}_{ij} \in \theta_{k,ij}} F(\mathbf{u}_{ij}) \leq \nu} bpa(\theta_{k,ij}) \quad (17.54)$$

The calculation of the partial belief can be found in Algorithm 2, line 6. Once the partial belief curve, for each coupled vector, is available, one can sample these curves, by taking a succession of $\{\nu_1, \dots, \nu^q, \dots, \nu_{N_S} = \nu\}$ values, and find the corresponding values of the coupled vectors $\hat{\mathbf{u}}_{k,ij}^q$. These values will be used in the next step to decouple the functions g_i (g_j) and compute the maxima of each g_i (g_j) with respect to the uncoupled variables \mathbf{u}_i (\mathbf{u}_j).

For each level q , given a fixed value of the coupling functions, one can study each g_i independently of the others. The idea is to run an optimisation for each function g_i over only the uncoupled vector \mathbf{u}_i . With the example in Fig. 17.1 in mind, having

$$\hat{h}_{ij}^q(\mathbf{u}_i) := h_{ij}(\tilde{\mathbf{d}}, \mathbf{u}_i, \hat{\mathbf{u}}_{ij}^q)$$

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

$$\begin{aligned} \hat{\mathbf{u}}_{k_1,1}^q &= \arg \max_{\mathbf{u}_1 \in \theta_{k_1,1}} g_1(\tilde{\mathbf{d}}, \mathbf{u}_1, \hat{h}_{12}^q(\mathbf{u}_1), \hat{h}_{13}^q(\mathbf{u}_1)), \forall \theta_{k_1,1} \subset \Theta_1 \\ \hat{\mathbf{u}}_{k_2,2}^q &= \arg \max_{\mathbf{u}_2 \in \theta_{k_2,2}} g_2(\tilde{\mathbf{d}}, \mathbf{u}_2, \hat{h}_{21}^q(\mathbf{u}_2), \hat{h}_{23}^q(\mathbf{u}_2)), \forall \theta_{k_2,2} \subset \Theta_2 \\ \hat{\mathbf{u}}_{k_3,3}^q &= \arg \max_{\mathbf{u}_3 \in \theta_{k_3,3}} g_3(\tilde{\mathbf{d}}, \mathbf{u}_3, \hat{h}_{31}^q(\mathbf{u}_3), \hat{h}_{32}^q(\mathbf{u}_3)), \forall \theta_{k_3,3} \subset \Theta_3 \end{aligned} \quad (17.55)$$

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

Once all the maxima over the focal elements of the uncoupled variables are available for each sample q one can calculate an approximation of $Bel(F(\mathbf{d}, \mathbf{u}) < \nu)$ as follows. From Eq. (17.55), for each sample q the maximum associated to the FE $\theta_k = \theta_{k_1,1} \times \theta_{k_2,2} \times \theta_{k_3,3}$, for $k = 1, \dots, N_{FE,1} \cdot N_{FE,2} \cdot N_{FE,3}$, given the condition of positive semidefinition of g_i , is

$$\max_{(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \in \theta_k} F(\tilde{\mathbf{d}}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \hat{\mathbf{u}}_{12}^q, \hat{\mathbf{u}}_{13}^q, \hat{\mathbf{u}}_{23}^q) = \hat{g}_{k_1,1}^q + \hat{g}_{k_2,2}^q + \hat{g}_{k_3,3}^q \quad (17.56)$$

with associated basic probability assignment:

$$bpa^q(\theta_k) = bpa(\theta_{k_1,1})bpa(\theta_{k_2,2})bpa(\theta_{k_3,3})\Delta Bel^q \quad (17.57)$$

where $\Delta Bel^q = \prod_{ij} \Delta Bel_{ij}^q$ are the contributions of the partial belief curves in (17.54): the generic ΔBel_{ij}^q is the difference of belief between sample q and $q - 1$ in the partial belief curve Bel_c about the coupled uncertain vector \mathbf{u}_{ij} . In other words, the bpa of each θ_k is the product of all the bpa 's of the FE of each uncoupled variable scaled with the product of the belief values of the samples drawn from the partial belief curves (Line 18). The approximation of the belief is then computed as:

$$\widetilde{Bel}(F(\mathbf{d}, \mathbf{u}) \leq \nu) = \sum_q \sum_k bpa^q(\theta_k) \quad (17.58)$$

17.4.4.3 Complexity Analysis

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

$$N_{FE} = \prod_{k=1}^m N_k. \quad (17.59)$$

In terms of coupled and uncoupled uncertain vectors we can write

$$N_{FE} = \left(\prod_{i=1}^{m_u} \prod_{k=1}^{p_i^u} N_{i,k}^u \right) \left(\prod_{i=1}^{m_c} \prod_{k=1}^{p_i^c} N_{i,k}^c \right) s \quad (17.60)$$

where p_i^u and p_i^c are the number of components of the i -th uncoupled and coupled vector, respectively, and $N_{i,k}^u$ and $N_{i,k}^c$ are the number of intervals of the k -th components of the i -th uncoupled and coupled vector, respectively.

The total number of FE that needs to be explored in the decomposition is instead:

$$N_{FE}^{Dec} = N_s \sum_{i=1}^{m_u} N_{FE,i}^u + \sum_{i=1}^{m_c} N_{FE,i}^c \quad (17.61)$$

considering the vector of uncertainties ordered as

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

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

17.4.5 Example

Some examples are proposed in this section to better clarify our approach to robust optimisation with particular emphasis to the decomposition approach used for uncertainty quantification within Dempster–Shafer Theory of Evidence (*DST*).

The multidisciplinary nature of a generic complex system is captured by the Evidence Network Model (*ENM*) described in Sect. 17.4.4.1 where the component's QoI contribute to the overall performance as stated in Eq.(17.46).

The analytical function $f = \sum_i g_i$ is here used as test case. The i -th node's performance and the generic coupling function h_{ij} between couples of nodes are, respectively, formulated as:

$$g_i = d_i + \sum_{k=1}^{N_{FE,i}^u} \sin u_{ik} - \left(\left| \sum_{k=1}^{N_{FE,i}^u} u_{ik} \right| + \sum_{k=1}^{m_c} h_{ik} \right)^3 \quad (17.62)$$

$$h_{ik} = \sum_{k=1}^{N_{FE,i}^u} \sin(u_{ik}) + u_{ik}$$

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

The scaling behaviour of the method is tested applying f to the different network topology represented in Fig. 17.4. In particular, each i -node depends on a pair of uncoupling uncertain variables $\mathbf{u}_{u,i} = [u_{u,i}^1, u_{u,i}^2]$ and each link shares between two nodes a pair of coupling uncertain variables $\mathbf{u}_{c,ij} = [u_{c,ij}^1, u_{c,ij}^2]$. Then, being n the number of nodes and l the number of links for a selected topology, the total number of uncertain variables is: $n_{u,tot} = 2(n + l)$.

The robust optimisation approach first evaluates the optimal design configuration \mathbf{d}_{opt} solving the constrained min-max optimisation problem described by Eq.(17.35) by means of Algorithm 1. Figure 17.3 presents, for example, the convergence of the algorithm for f applied to topology (c) where the constraint function has been defined as:

$$C(\mathbf{d}, \mathbf{u}) = - \sum_i d_i + 5 \sum_i (\sin(d_i u_i)) \leq 0 \quad (17.63)$$

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

$$f_{max} = \max_{\mathbf{u} \in U} f(\mathbf{d}, \mathbf{u}) \quad (17.64)$$

and the constraint function

$$c_{max} = \max_{\mathbf{u} \in U} c(\mathbf{d}, \mathbf{u}) \leq 0 \quad (17.65)$$

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

The network' topology is then exploited with Algorithm 2 to propagate epistemic uncertainty and calculate at a reduced computational cost the Belief and Plausibility curves corresponding to \mathbf{d}_{opt} . For this purpose, the Frame of Discernment as defined in Sect. 17.2.1 is constructed assigning to each uncertain variable u two possible intervals and their bpa . The total number of focal elements is then $n_{fe} = 2^{n_{u,tot}}$. The number of optimisations required for exact quantification of the Belief curve is $n_{opt,exact} = n_{fe}$ while the number required by the decomposition approach follows Eq. (17.60).

Table 17.1 collects the results of the simulations for the different network' topology. It is shown the gain in computational cost offered by the decomposition approach together with the generated error. The error has been evaluated as the ratio $(A_{exact} - A_{dec})/A_{dec}$ where A_{exact} and A_{dec} are the integral of the exact and decomposition curve, respectively. For problems satisfying the conditions in Sect. 17.4.4.1, the decomposition approach assures to quantify exactly the DST' measures of probability when the partial curves are entirely sampled. For example, considering the topology (c) and using 64 samples (all the combinations for the 4 samples for each of the 3 coupled curve), we obtain an error equal to zero running a number of optimisations that is 19% of total number of focal elements. Furthermore, the smaller is the number of samples, the lower is the computational cost but the higher the error. For example, for the same problem, a single sample brings to a cost that is 0.59% of the exact evaluation increasing the error, however, to 187%. Between these two extreme positions we can make a trade-off between cost and accuracy.

Figure 17.2 shows instead the plots of the partial curves and the final curves calculated with the decomposition strategy and also the exact curves calculated running an optimisation for each focal element for f applied to topology (c).

It has been noticed that sampling very close points in the space of the maxima in Fig. 17.2a brings to negligible contribution to the error reduction in Fig. 17.2b. Then, future works will show how a careful selection of the samples can improve the algorithm performance mitigating both the error (possibly bringing it to zero) and the cost under the condition that the user is not interested in the entire curve but only in the value assumed by belief and plausibility at some specific thresholds. This is particularly important when the decomposition strategy is nested within an optimisation loop, for example, in the approach presented in Eq. (17.39).

17.5 Conclusions

In this chapter we described the importance of uncertainty quantification in real world modern problems and its challenge due to the computational cost, an overview of the uncertain probability theory and a deeper insight in Evidence Theory. Two novel tools have been described to do a rigorous uncertainty quantification when the

Table 17.1 ENM decomposition results

Topology	n_{fe}	$n_{samp,i}$	$n_{samp,tot}$	n_{opt}	n_{opt}/n_{fe}	CPU f (s)	CPU fmincon (s)	CPU Alg. (s)	Belief error
(a)	$2^6 = 64$	1	1	12	0.1875	0.008	0.341	0.4265	0.46635
		2	2	20	0.3125	0.01	0.312	0.4685	0.40311
		3	3	28	0.4375	0.018	0.377	0.5570	0.20439
		4	4	36	0.5625	0.019	0.448	0.6415	$2e-7$
(b)	$2^{10} = 1024$	1	1	20	0.0195	0.014	0.387	0.662	1.524
		2	4	56	0.0546	0.031	0.692	0.930	4.344
		3	9	116	0.1133	0.053	1.137	1.435	0.1618
		4	16	200	0.1953	0.089	1.896	2.288	0.0001
(c)	$2^{12} = 4096$	1	1	24	0.0059	0.017	0.428	0.732	1.8705
		2	8	108	0.0264	0.0525	1.061	1.353	0.59611
		3	27	336	0.0820	0.151	2.95	3.492	0.14550
		4	64	780	0.1904	0.346	6.62	7.650	0.00004
(d)	$2^{22} = 4,194,304$	1	1	44	1.049e-05	0.0495	0.694	1.17	2.6101
		2	32	788	1.878e-04	0.611	7.202	10.717	0.5776
		3	243	5852	1.395e-03	4.32	48.50	73.38	0.1870
		4	1024	24,596	5.864e-03	17.88	200.28	299.21	0.0002

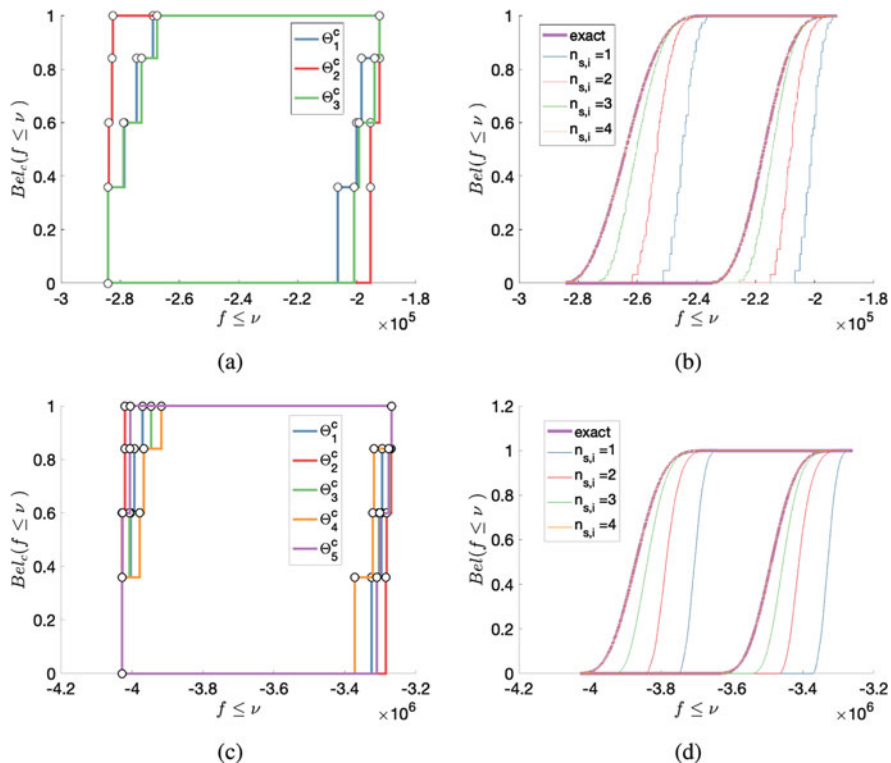
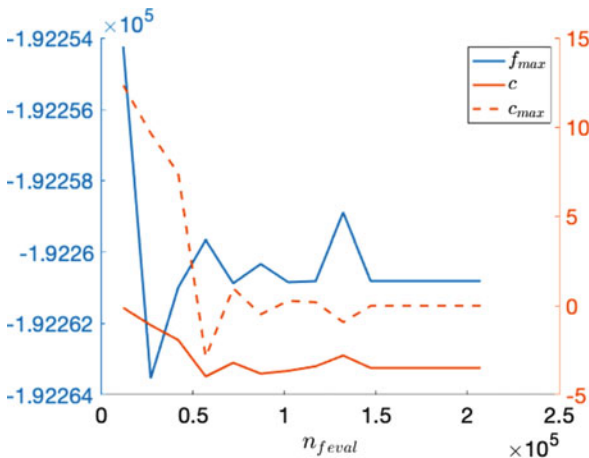


Fig. 17.2 Belief and Plausibility curves for f . Sub-figures (a, b) refer to topology (c), while sub-figures (c, d) refer to topology (d). Sub-figures (a, c) plot the partial curves evaluated only in the subspace of the coupling uncertain variables. Each colour corresponds to a single link in the network. Sub-figures (b, d) show the final curves calculated with the decomposition approach where each colour refers to a different sampling. They show also the exact belief and plausibility evaluated running an optimisation for each focal element

Fig. 17.3 Convergence to the optimal solution of the constrained min-max problem for Algorithm 1 with f applied to topology (c). For each design solution proposed by the algorithm at each new iteration, it is here plotted the worst case scenario in the uncertain space for the objective function f_{max} and for the constraint violation c_{max} . It is also plotted the value of the constraint c corresponding to the worst scenario for f_{max}



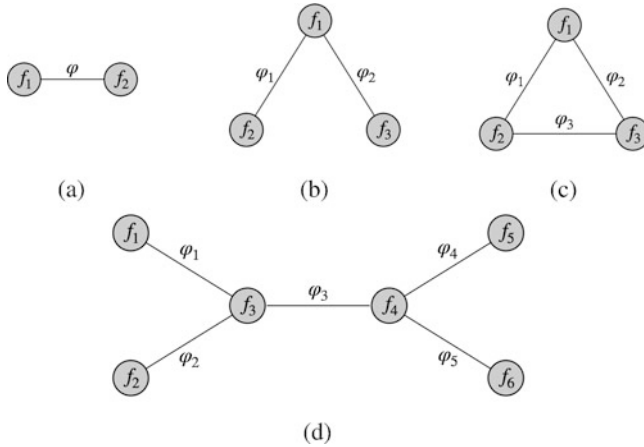


Fig. 17.4 Network topology applied to TC_1 and TC_2 for the study of the scalability of the decomposition method. (a) Simple graph with dimension of the uncertain space $\dim_u = 6$. (b) Triad \wedge with $\dim_u = 10$. (c) Triangle \mathbb{K}_3 with $\dim_u = 12$. (d) Graph with $\dim_u = 22$

problem is affected by lack of knowledge and described by Imprecise Probability Theories in order to overcome the limits imposed by the widely used contingencies and margins approach. The min-max algorithm [78] has been presented to do a conservative and robust optimisation under uncertainty and finally a novel Evidence Network Model [90, 91] has been described to decompose a complex system and propagate uncertainty to evaluate the cumulative Belief and Plausibility curves.

References

1. V. Larouche, NASA mass growth analysis - spacecraft & subsystems, in *2014 NASA Cost SymposiumLaRC, August 14*, vol. 117 (2014)
2. S. Division, Space engineering, Engineering design model data exchange (CDF) (2010)
3. ANSI/AIAA, S-120A-201X Draft for public review American national standard mass properties control for space systems (2015)
4. D. Woods, Creating foresight: how resilience engineering can transform NASAs approach to risky decision making. *Work* **4**, 137–144 (2003)
5. A.M. Madni, S. Jackson, Towards a conceptual framework for resilience engineering. *IEEE Syst. J.* **3**(2), 181–191 (2009). <https://doi.org/10.1109/JSYST.2009.2017397>
6. A.W. Wymore, *Model-Based Systems Engineering* (U. of A. series editor A. Terry Bahill, Ed.) (C. Press, Boca Raton, Florida, 1993)
7. S.A. Sheard, Twelve Systems Engineering Roles. *INCOSE Int. Symp.* **6**, 478–485 (2014). <https://doi.org/10.1002/j.2334-5837.1996.tb02042.x>
8. N.V. Sahinidis, Optimization under uncertainty: state-of-the-art and opportunities. *Comput. Chem. Eng.* **28**, 97–183 (2004). <https://doi.org/10.1016/j.compchemeng.2003.09.017>
9. T.A. Zang, M.J. Hemsch, M.W. Hilburger, S.P. Kenny, J.M. Luckring, P. Maghami, et al., Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles. NASA Tech Reports Serv 211462:paste (2002)

10. H.G. Beyer, B. Sendhoff, Robust optimization - a comprehensive survey. *Comput. Methods Appl. Mech. Eng.* **196**, 3190–3218 (2007). <https://doi.org/10.1016/j.cma.2007.03.003>
11. E. Zio, Reliability engineering: old problems and new challenges. *Reliab. Eng. Syst. Saf.* **94**, 125–141 (2009). <https://doi.org/10.1016/j.ress.2008.06.002>
12. G. Punzo, A. Tewari, E. Butani, M. Vasile, A. Purvis, M. Mayfield, et al., Engineering resilient complex systems: the necessary shift toward complexity science. *IEEE Syst. J.* **14**, 3865–3874 (2020). <https://doi.org/10.1109/jsyst.2019.2958829>
13. S.N. Naghshbandi, L. Varga, A. Purvis, R. McWilliam, E. Minisci, M. Vasile, et al., A review of methods to study resilience of complex engineering and engineered systems. *IEEE Access* **44**, 11 (2020). <https://doi.org/10.1109/access.2020.2992239>
14. C.N. Calvano, P. John, Systems engineering in an age of complexity. *Syst. Eng.* **7**, 25–34 (2004). <https://doi.org/10.1002/sys.10054>
15. S.A. Sheard, A. Mostashari, Principles of complex systems for systems engineering. *Syst. Eng.* **12**, 295–311 (2009). <https://doi.org/10.1002/sys.20124>
16. W. Yao, X. Chen, W. Luo, M. van Tooren, J. Guo, Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. *Prog. Aerosp. Sci.* **47**, 450–479 (2011). <https://doi.org/10.1016/J.PAEROSCI.2011.05.001>
17. J.C. Helton, J.D. Johnson, W.L. Oberkampf, C.J. Sallaberry, Representation of analysis results involving aleatory and epistemic uncertainty. *Int. J. Gen. Syst.* **39**, 605–646 (2010). <https://doi.org/10.1080/03081079.2010.486664>
18. A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Statist.* **38**(2), 325–339 (1967)
19. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)
20. M. Vasile, E. Minisci, F. Zuiani, E. Komninou, Q. Wijnands, Fast evidence-based space system engineering, in *IAC* (2011)
21. C.O. Absil, G. Filippi, A. Riccardi, M. Vasile, A variance-based estimation of the resilience indices in the preliminary design optimisation of engineering systems under epistemic uncertainty, in *EUROGEN* (2017)
22. J.C. Helton, J.D. Johnson, W.L. Oberkampf, C.B. Storlie, A sampling-based computational strategy for the representation of epistemic uncertainty in model predictions with evidence theory. *Comput. Methods Appl. Mech. Eng.* **196**, 3980–3998 (2007). <https://doi.org/10.1016/j.cma.2006.10.049>
23. W. Yao, X. Chen, Y. Huang, Z. Gurdal, M. Van Tooren, Sequential optimization and mixed uncertainty analysis method for reliability-based optimization. *AIAA J.* **51**, 2266–2277, (2013). <https://doi.org/10.2514/1.J052327>
24. H. Agarwal, J.E. Renaud, E.L. Preston, D. Padmanabhan, Uncertainty quantification using evidence theory in multidisciplinary design optimization. *Reliab. Eng. Syst. Saf.* **85**, 28194 (2004). <https://doi.org/10.1016/J.RESS.2004.03.017>
25. Z.P. Mourelatos, J. Zhou, A design optimization method using evidence theory. *J. Mech. Des. Trans. ASME* **128**, 901–908 (2006). <https://doi.org/10.1115/1.2204970>
26. P. Pedersen, C.L. Laureen, Design for minimum stress concentration by finite elements and linear programming. *J. Struct. Mech.* **10**, 375–391 (1982). <https://doi.org/10.1080/03601218208907419>
27. M. Nicolich, G. Cassio, System models simulation process management and collaborative multidisciplinary optimization, in *CEUR Workshop Proceedings, Rome*, vol. 1300 (2014), pp. 1–16
28. G.J. Park, T.H. Lee, K.H. Lee, K.H. Hwang, Robust design: an overview. *AIAA J.* **44**, 181–191 (2006). <https://doi.org/10.2514/1.13639>
29. M. Kalsi, K. Hacker, K. Lewis, A comprehensive robust design approach for decision trade-offs in complex systems design. *J. Mech. Des. Trans. ASME* **123**, 1–10 (2001). <https://doi.org/10.1115/1.1334596>
30. X. Du, W. Chen, Towards a better understanding of modeling feasibility robustness in engineering design. *J. Mech. Des. Trans. ASME* **122**, 385–394 (2000). <https://doi.org/10.1115/1.1290247>

31. X. Zhuang, R. Pan, L. Wang, Robustness and reliability consideration in product design optimization under uncertainty, in *IEEE International Conference on Industrial Engineering and Engineering Management* (IEEE, Piscataway, 2011), pp. 132–159. <https://doi.org/10.1109/IEEM.2011.6118131>
32. A. Lewis, Robust regularization, Technical Report, Simon Fraser University, Vancouver (2002)
33. M. Trosset, Taguchi and robust optimization, Technical Report, 96-31, Department of Computational & Applied Mathematics, Rice University (1996)
34. M. McIlhagga, P. Husbands, R. Ives, A comparison of search techniques on a wing-box optimisation problem, in *Parallel Problem Solving from Nature* ed. by H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel, vol. 4 (Springer, Berlin, 1996), pp. 614–623
35. J. Herrmann, A genetic algorithm for minimax optimization problems, in *Proceedings of the Congress on Evolutionary Computation*, vol. 2 (IEEE Press, New York, 1999), pp. 1099–1103
36. L. El Ghaoui, H. Lebret, Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.* **18**(4), 1035–1064 (1997)
37. H.-G. Beyer, M. Olhofer, B. Sendhoff, On the behavior of μ/μ_1 , λ -ES optimizing functions disturbed by generalized noise, in *Foundations of Genetic Algorithms*, ed. by K. De Jong, R. Poli, J. Rowe, vol. 7 (Morgan Kaufman, San Francisco, 2003), pp. 307–328
38. I. Das, Robustness optimization for constrained, nonlinear programming problems, Tech. Rep. TR97-01, Technical Reports, Department of Computational & Applied Mathematics, Rice University, Houston, TX (1997)
39. J. Mulvey, R. Vanderbei, S. Zenios, Robust optimization of large-scale systems. *Oper. Res.* **43**(2), 264–281 (1995)
40. W. Chen, M. Wiecek, J. Zhang, Quality utility a compromise programming approach to robust design. *ASME J. Mech. Des.* **121**(2), 179–187 (1999)
41. N. Rolander, J. Rambo, Y. Joshi, J. Allen, F. Mistree, An approach to robust design of turbulent convective systems. *J. Mech. Des.* **128**(4), 844–855 (2006)
42. Y. Jin, B. Sendhoff, Trade-off between performance and robustness: an evolutionary multiobjective approach, in *Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003*, ed. by C. Fonseca, P. Fleming, E. Zitzler, K. Deb (Springer, Heidelberg, 2003), pp. 237–251
43. W.L. Oberkampf, J.C. Helton, Investigation of evidence theory for engineering applications, in *Fourth Non-Deterministic Approaches Forum*, vol. 1569 (AIAA, Reston, 2002)
44. L. Simoes, Fuzzy optimization of structures by the two-phase method. *Comput. Struct.* **79**(2628), 2481–2490 (2001)
45. F. Campos, J. Villar, M. Jimenez, Robust solutions using fuzzy chance constraints. *Eng. Optim.* **38**(6), 627–645 (2006)
46. J. Liu, J.-B. Yang, J. Wang, H.S. Sii, Review of uncertainty reasoning approaches as guidance for maritime and offshore safety-based assessment. *J. UK Saf. Reliab. Soc.* **23**(1), 63–80 (2002)
47. M. Vasile, Robust mission design through evidence theory and multiagent collaborative search. *Ann. New York Acad. Sci.* **1065**, 152–173 (2005)
48. N. Croisard, M. Vasile, S. Kemble, G. Radice, Preliminary space mission design under uncertainty. *Acta Astronaut.* **66**, 5–6 (2010)
49. R.W. Chaney, A method of centers algorithm for certain minimax problems. *Math. Program.* **22**(1), 202–226 (1982)
50. R. Klessig, E. Polak, A method of feasible directions using function approximations, with applications to min max problems. *J. Math. Anal. Appl.* **41**(3) 583–602 (1973)
51. V. Panin, Linearization method for continuous min-max problem. *Cybernetics* **17**(2), 239–243 (1981)
52. Y. Danilin, V. Panin, B. Pshenichnyi, On the Shannon Gapacity of a graph. *Notes Control Inf. Sci.* **23**(30), 51–57 (1982)
53. V.F. Damyranov, *VN Malozemov* (Wiley, New York, 1974)

54. D. Agnew, Improved minimax optimization for circuit design. *IEEE Trans. Circuits Syst.* **28**(8), 791–803 (1981)
55. J. Shinn-Hwa Wang, W. Wei-Ming Dai, Transformation of min-max optimization to least-square estimation and application to interconnect design optimization, in *Proceedings of ICCD'95 International Conference on Computer Design. VLSI in Computers and Processors* (IEEE Computer Society Press, Washington, 1995), pp. 664–670
56. B. Lu, Y. Cao, M. Yuan, J. Zhou, Reference variable methods of solving minimax optimization problems. *J. Global Optim.* **42**(1), 1–21 (2008)
57. M. Sainz, P. Herrero, J. Armengol, J. Veh, Continuous minimax optimization using modal intervals. *J. Math. Anal. Appl.* **339**, 18–30 (2008)
58. Y. Feng, L. Hongwei, Z. Shuisheng, L. Sanyang, A smoothing trust-region Newton-CG method for minimax problem. *Appl. Math. Comput.* **199**(2), 581–589 (2008)
59. P. Parpas, B. Rustem, An algorithm for the global optimization of a class of continuous minimax problems. *J. Optim. Theory Appl.* **141**, 46–173 (2009). <https://doi.org/10.1007/s10957-008-9473-4>
60. H. Aissi, C. Bazgan, D. Vanderpooten, Min-max and min-max regret versions of combinatorial optimization problems: a survey. *Eur. J. Oper. Res.* **197**, 427–438 (2009)
61. T.M. Cavalier, W.A. Conner, E. del Castillo, S.I. Brown, A heuristic algorithm for minimax sensor location in the plane. *Eur. J. Oper. Res.* **183**(1), 42–55 (2007)
62. D. Ahr, G. Reinelt, A tabu search algorithm for the min-max k-Chinese postman problem. *Comput. Oper. Res.* **33**(12), 3403–3422 (2006)
63. A.M. Cramer, S.D. Sudhoff, E.L. Zivi, Evolutionary algorithms for minimax problems in robust design. *IEEE Trans. Evol. Comput.* **13**(2), 444–453 (2009)
64. R.I. Lung, D. Dumitrescu, A new evolutionary approach to minimax problems, in *IEEE Congress on Evolutionary Computation (CEC)*, 5–8 June 2011, New Orleans (2011), pp. 1902–1905. <https://doi.org/10.1109/CEC.2011.5949847>
65. A. Zhou, Q. Zhang, A surrogate-assisted evolutionary algorithm for minimax optimization, in *IEEE Conference on Evolutionary Computation (CEC)* (2010)
66. E.C. Laskari, K.E. Parsopoulos, M.N. Vrahatis, Particle swarm optimization for minimax problems, in *Proceedings of the 2002 Congress on Evolutionary Computation* (IEEE Press, New York, 2002), pp. 1582–1587
67. W. Conner, Comparison of evolutionary algorithms on the minimax sensor location problem. *The Pennsylvania State University*, 310.
68. A.M. Cramer, S.D. Sudhoff, E.L. Zivi, Evolutionary algorithms for minimax problems in robust design. *IEEE Trans. Evol. Comput.* **13**, 444–453 (2009). <https://doi.org/10.1109/TEVC.2008.2004422>
69. D. Agnew, Improved minimax optimization for circuit design. *IEEE Trans. Circuits Syst.* **28**(8), 791–803 (1981)
70. A.V. Sebal, J. Schlenzig, Minimax design of neural net controllers for highly uncertain plants. *IEEE Trans. Neural Netw.* **5**(1), 73–82 (1994)
71. H.J.C. Barbosa, A coevolutionary genetic algorithm for a game approach to structural optimization, in *Proceedings of the 7-th International Conference on Genetic Algorithms* (1997), pp. 545–552
72. H.J.C. Barbosa, A coevolutionary genetic algorithm for constrained optimization, in *Proceedings of 1999 Congress on Evolutionary Computation*, ed. by P. Angeline et. al. (1997), pp. 1605–1611
73. J.W. Herrmann, A genetic algorithm for minimax optimization problems, in *Proceedings of 1999 Congress on Evolutionary Computation*, ed. by P. Angeline et. al. (1997), pp. 1099–1103
74. T.M. Jensen, A new look at solving minimax problems with coevolutionary genetic algorithms, in *Metaheuristics: Computer Decision-Making*. Applied Optimization, vol. 86 (Springer, Boston, 2003), pp 369–384
75. Y.S. Ong, P.B. Nair, A.J. Keane, K.W. Wong, Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems, in *Knowledge Incorporation in Evolutionary Computation* (Springer, Berlin, 2004), pp. 307–332

76. Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput.* **9**(1), 3–12 (2005)
77. J. Marzat, E. Walker, H. Piet-Lahanier, Worst-case global optimization of black-box functions through kriging and relaxation. *J. Global Optim.* **55**, 707–727 (2013)
78. G. Filippi, M. Vasile, A memetic approach to the solution of constrained min-max problems, in *2019 IEEE Congress on Evolutionary Computation (CEC)* (2019), pp. 506–513. <https://doi.org/10.1109/CEC.2019.8790124>
79. J.C. Helton, J. Johnson, W.L. Oberkampf, C. Sallaberry, Sensitivity analysis in conjunction with evidence theory representations of epistemic uncertainty. *Reliab. Eng. Syst. Saf.* **91**(10–11), 1414–1434 (2006)
80. C. Joslyn, J.C. Helton, Bounds on belief and plausibility of functionality propagated random sets, in *2002 Annual Meetings of the North American Fuzzy Information Processing Society, Proceedings*, ed. by J. Keller, O. Nasraoui, June 2002, New Orleans, vol. 2729 (IEEE, Piscataway, 2002), pp. 412–417
81. C. Joslyn, V. Kreinovich, Convergence properties of an interval probabilistic approach to system reliability estimation. *Int. J. Gen. Syst.* **34**(4), 465–482 (2005)
82. M. Di Carlo, M. Vasile, C. Greco, R. Epenoy, Robust optimisation of low-thrust interplanetary transfers using evidence theory, in *29th AAS/AIAA Space Flight Mechanics Meeting* (2019)
83. F. Voorbraak, A computationally efficient approximation of Dempster-Shafer theory. *Int. J. Man-Mach. Stud.* **30**, 525–536 (1989)
84. D. Dubois, H. Prade, Consonant approximations of belief functions. *Int. J. Approx. Reason.* **4**, 419–449 (1990)
85. B. Tessem, Approximations for efficient computation in the theory of evidence. *Artif. Intell.* **61**(2), 315–329 (1993)
86. M. Bauer, Approximations for decision making in the Dempster-Shafer theory of evidence, in *Proceedings of Twelfth International Conference on Uncertainty in Artificial Intelligence* (1996), pp. 73–80
87. D. Han, J. Dezert, Y. Yang, Two novel methods for BBA approximation based on focal element redundancy, in *18th International Conference on Information Fusion* (2015), p. 428–434
88. T. Denux, Inner and outer approximation of belief structures using a hierarchical clustering approach. *Int. J. Uncertainty Fuzziness Knowledge Based Syst.* **9**(4), 437–460 (2001)
89. D. Harmanec, Faithful approximations of belief functions, in *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence* (1999), p. 2718
90. M. Vasile, G. Filippi, C. Ortega, A. Riccardi, Fast belief estimation in evidence network models, in *EUROGEN 2017* (2017)
91. S. Alicino, M. Vasile, Evidence-based preliminary design of spacecraft, in *6th International Conference on Systems & Concurrent Engineering for Space Applications SECESA 2014* (2014)
92. G. Filippi, M. Vasile, D. Krpelik, P.Z. Korondi, M. Marchi, C. Poloni, Space systems resilience optimisation under epistemic uncertainty. *Acta Astronaut.* **165**, 195–210 (2019). <https://doi.org/10.1016/j.actaastro.2019.08.024>