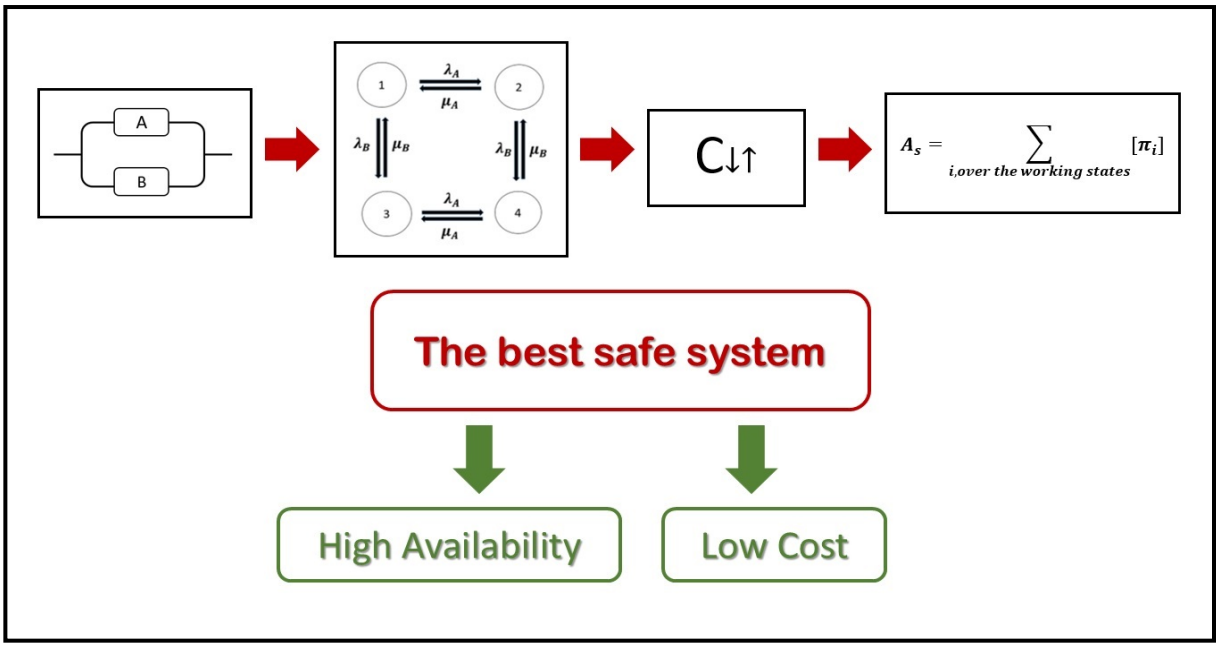
Optimization of the availability of multi-states systems under uncertainty

Joanna Akrouche

Par **Joanna AKROUCHE**

Optimization of the availability of multi-states systems under uncertainty

Thèse présentée en cotutelle
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 2 mars 2020
Spécialité : Sciences et Technologies de l'Information et des Systèmes
: Unité de recherche Heudyasic (UMR-7253)

D2545

Optimization of the Availability of Multi-States Systems under Uncertainty

Joanna AKROUCHE

Spécialité : Sciences et Technologies de l'Information et des Systèmes

A thesis presented for the degree of Doctor

02 March 2020

Université de Technologie de Compiègne
Lebanese University
Heudiasyc Laboratory, UMR UTC/CNRS 7253

Jury Members

Mrs. Farah CHEHADE, MCF-HDR, Université de Technologie de Troyes, Reporter Mr. Mustapha NOUR EL FATH, Professor, Université Laval, Reporter
Mr. Jacques PELLETAN, MCF, Université Paris 8, Examiner
Mr. Amadou GNING, Lecturer, University of Hull, Examiner
Mr. Walter SCHÖN, Professor, Université de Technologie de Compiègne, Examiner Mr. Mohamed SALLAK, MCF-HDR, Université de Technologie de Compiègne, Director Mr. Fahed ABDALLAH, Professor, Lebanese University, Director (LU)
Mr. Eric CHÂTELET, Professor, Université de Technologie de Troyes, Supervisor
Mrs. Hiba Haj CHHADÉ, EC, Lebanese University, Supervisor (invited)

Optimisation de la Disponibilité d'un Système Multi-États en présence d'Incertitudes

Joanna AKROUCHE

Thèse présentée pour l'obtention du grade de Docteur

02 Mars 2020

Université de Technologie de Compiègne
Université Libanaise
Heudiasyc Laboratory, UMR UTC/CNRS 7253

Jury Members

Mme. Farah CHEHADE, MCF-HDR, Université de Technologie de Troyes, Rapporteur
M. Mustapha NOUR EL FATH, Professeur, Université Laval, Rapporteur
M. Jacques PELLETAN, MCF, Université Paris 8, Examinateur
M. Amadou GNING, Lecturer, University of Hull, Examinateur
M. Walter SCHÖN, Professeur, Université de Technologie de Compiègne, Examinateur
M. Mohamed SALLAK, MCF-HDR, Université de Technologie de Compiègne, Directeur
M. Fahed ABDALLAH, Professeur, Université Libanaise, Directeur (UL)
M. Eric CHÂTELET, Professor, Université de Technologie de Troyes, Encadrant
Mme. Hiba Haj CHHADÉ, EC, Université Libanaise, Encadrante (invitée)

Remerciements

Cette thèse fait partie d'un accord de thèse cotutelle entre l'École Doctorale des Sciences et Technologies de l'Université Libanaise et le Laboratoire Heudiasyc UMR CNRS 7253 de l'Université de Technologie de Compié gne, et a été réalisée dans l'équipe SCOP (Sûreté, Communication, Optimisation). Je tiens d'abord à remercier les deux établissements à me donner l'opportunité de réaliser cette thèse dans les meilleures conditions.

J'exprime également ma grande gratitude envers l'Univérsité Libanaise et la région Hauts-de-France pour le financement pendant la période de thèse.

Je tiens à remercier chaleureusement mes deux directeurs de thèse, Mohamed SALLAK et Fahed ABDALLAH, qui m'ont accompagné et soutenu tout au long de ce travail. Leurs précieux conseils n'ont cessé d'aiguiller et d'approfondir ma réflexion.

Je remercie également mes encadrents Eric CHÂTELET et Hiba HAJ CHHADÉ pour leur disponibilité et leurs qualités professionnelles qui m'ont énormément aidé dans la réalisation de ce travail.

Je remercie trés vivement tous les membres du jury de cette thèse: Farah CHEHADE, Mustapha NOUR EL FATH, Jacques PELLETAN, Amadou Gning et Walter SCHÖN, qui m'ont honoré en acceptant de la lire, de l'évaluer et de la juger.

J'exprime ma reconnaissance à l'égard de toutes les personnes que j'ai rencontrées au cours de ce travail et qui, par nos discussions et leurs idées, m'ont aidée à le construire.

Travailler côte à côte, pendant des années, tisser des liens et des amitiés, pour cela je remercie tous mes collègues au laboratoire et surtout mes collègues au bureau: Rim LOUHICHI, Xuhong LI et Ariane SPAENLEHAUER qui sont devenus des amis pas juste des collègues.

Je voudrais associer à ces remerciements tous mes précieux amis, Josephine MERHI BLEIK, Hani AL HAJJAR, Sam TAOUM et Youssef RAHMÉ, qui ont contribué de près ou de loin à la réussite de ma thèse de doctorat, et qui ont été toujours à côté de moi moralement et physiquement.

Enfin, un immense merci à tous mes proches pour leur soutien indéfectible, surtout à mes parents Michel et Aurore, mon frère George, mes deux soeurs Jocelyne et Gisèlle à qui je dois d'avoir mené à bien ce travail.

Publications

1. **International conferences**

   - J. M. Akrouche, M. Sallak, E. Châtelet, F. A. Abdallah et H. Z. Haj Chhadé. A Contribution to the Evaluation of Imprecise Availability of Complex Systems Using Markov Models. 2nd ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering, Greece, 2017.
   - J. Akrouche, M. Sallak, E. Châtelet, F. Abdallah et H. Haj Chhadé. New method for availability computing of complex systems using Imprecise Markov models, 10th IMA International Conference on Modelling in Industrial Maintenance and Reliability, England, 2018.
   - J. Akrouche, M. Sallak, E. Châtelet, F. Abdallah et H. Haj Chhadé. Methodology for Imprecise Availability Computing and Optimization, 29th European Safety and Reliability Conference (ESREL), Germany, 2019.

2. **Submitted journal articles**

   - J. Akrouche, M. Sallak, E. Châtelet, F. Abdallah et H. Haj Chhadé. Methodology for the assessment of Imprecise Multi-State System Availability, IEEE Transactions on Reliability.
   - J. Akrouche, M. Sallak, E. Châtelet, F. Abdallah et H. Haj Chhadé. Optimization of the Imprecise Availability of Multi-State Systems, for the Special Issue on Non-probabilistic and hybrid approaches for uncertainty quantification and reliability analysis in the ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering (where the abstract was accepted).

3. **Preparation of a journal article**

   - J. Akrouche, M. Sallak, E. Châtelet, F. Abdallah et H. Haj Chhadé. A point of view for optimization of the Imprecise Availability of Multi-State Systems.

Abstract

Dependability has become a necessity in the industrial world during the twentieth century. Dependability is an activity domain that proposes means to increase the attributes of the system in a reasonable time and with a less cost.

In systems engineering, dependability is defined as the property that enables system users to place a justified confidence in the service it delivers to them and it is a measure of a system's availability, reliability, and its maintainability, and maintenance support performance, and, in some cases, other characteristics such as durability, safety and security. The key concept that our work is based on is the **availability**. The availability $A(t)$ is the ability of a system to be operational at a specific moment.

The cost of some system with high availability is very expensive. The designer must compromise between the availability and the economic costs. Users can reject systems that are unsafe, unreliable or insecure. Therefore, any user (or industry) will ask this question before getting any product: "What is the optimal product in the market?" To answer to this question, we must combine the following two points:

- The best availability of the system: The user wants a product that lasts as long as possible.
- The best cost of the system: The user wants a product without costing him a fortune.

Availability calculation is based primarily on knowledge of failure rates and repairs of system components. Availability analysis helps to calculate the ability of a system to provide a required level of performance depending on the level of degradation.

Several methods have been used to calculate the availability of a system, amongst which we find the Universal Generating Function (UGF), Inclusion-Exclusion technique, Markov models, etc. These methods employ different probabilistic techniques to evaluate this criterion, but these proposed approaches remain effective only for very specific cases, for exam-

ple the cases of binary systems.

A binary system is a system where only two cases are possible: perfect functioning and total failure. While the transition to multi-state systems (MSS) drastically restricts the application of most of these methods. In real life, the systems corresponds to MSS. In such scenarios, systems and their components can operate at different performance levels between working and failure states. However, the evaluation of the availability of the MSSs is more difficult than in the binary case, because we have to take into account the different combinations of the component failure modes. Throughout this thesis, we search for a method that helps us to compute and to optimize the availability of MSS.

Résumé

La sûreté de fonctionnement (SdF) est devenue une nécessité dans le monde industriel au cours du XXe siècle. La SdF est un domaine d'activité qui propose des moyens d'augmenter les attributs du système dans un délai raisonnable et à moindre coût.

Dans l'ingénierie des systèmes, la SdF est définie comme la propriété qui permet aux utilisateurs du système de placer une confiance justifiée dans le service qu'il leur fournit et c'est une mesure de la disponibilité, de la fiabilité et de la maintenabilité d'un système, et de la performance du support de maintenance, et, dans certains cas, d'autres caractéristiques telles que la durabilité, la sûreté et la sécurité. Le concept sur lequel notre travail est basé est la textbf disponibilité. La disponibilité $A(t)$ est la capacité d'un système à être opérationnel à un moment précis.

Le coût d'un système à haute disponibilité est très cher. Le concepteur doit faire un compromis entre la disponibilité et les coûts économiques. Les utilisateurs peuvent rejeter des systèmes dangereux, peu fiables ou non sécurisés. Par conséquent, tout utilisateur (ou industrie) posera cette question avant avoir un produit: "Quel est le produit optimal sur le marché?" Pour répondre à cette question, nous devons combiner les deux points suivants:

- La meilleure disponibilité du système: L'utilisateur souhaite un produit qui dure longtemps le plus possible.
- Le meilleur coût du système: l'utilisateur veut un produit sans lui coûter une fortune.

Le calcul de la disponibilité est basé principalement sur la connaissance des taux de défaillance et des réparations des composants du système. L'analyse de disponibilité permet de calculer la capacité d'un système à fournir un niveau de performance requis en fonction du niveau de dégradation.

Plusieurs méthodes ont été utilisées pour calculer la disponibilité d'un système, parmi lesquelles on trouve la Fonction de Génératrice Universelle (UGF), la technique d'inclusion-

exclusion, les modèles de Markov, etc. Ces méthodes utilisent différentes techniques probabilistes pour évaluer ce critère, mais ces approches proposées ne restent efficaces que pour des cas très spécifiques, par exemple les cas de systèmes binaires.

Un système binaire est un système où deux cas sont possibles: fonctionnement parfait et défaillance totale. Alors que les systèmes multi-états (SME) restreint considérablement l'application de la plupart de ces méthodes. Dans la vie réelle, les systèmes correspondent à des SME. Dans de tels scénarios, les systèmes et leurs composants peuvent fonctionner à différents niveaux de performances entre l'etat de fonctionnement parfait et l'état de défaillance totale. Cependant, l'évaluation de la disponibilité des SME est plus difficile que dans le cas binaire, car il faut tenir compte des différentes combinaisons des modes de défaillance des composants. Tout au long de cette thèse, nous recherchons une méthode qui nous aide à calculer et à optimiser la disponibilité de SME tenant compte le facteur coût.

List of Figures

List of Tables

List of Acronyms

UGF	Universal Generating Function.
RAMS	Reliability, Availability, Maintainability, and Safety.
MSS	Multi-states system.
PHA	Process Hazard Analysis.
FT	Fault Tree.
MTTF	Mean time to failure.
MTBF	Mean time between consecutive failures
MTTR	Mean time to repair.
FBP	Forward-Backward propagation.
R	Reliability.
A	Availability.
F	Unreliability.
M	Maintainability.
MUT	Mean Up Time.
MDT	Mean Down Time.
FMEA	Failure Modes and Effects Analysis.
RBD	Reliability Block Diagram.
TTM	Truth-Table method.
FTA	Fault Tree Analysis.
ACT	Accident Consequence Tree.
CCD	Cause-Consequence Diagram Method.

BN	Bayesian Network.
CPT	Conditional Probability Table.
MSBN	Microsoft Bayesian Network Editor.
CSP H	Constraint satisfaction problem.
$\mathbb{S}$	The solution set of H.
C	Contractor.
C_{GE}	Contractor of Gauss elimination.
C_{GS}	Contractor of Gauss-Siedel.
C_K	Krawczyk contrator.
C_N	Newton contractor.
$C_{\downarrow\uparrow}$	Forward-backward propagation.
S	The system.
E	Space of states.
Q	Transition matrix.
PMP	Primary Minimum Path.
SP	Secondary Path.
IUGF	Interval Universal Generating Function.
BUGF	Belief Universal Generating Function.
Pl	Plausibility.
Bel	Belief.
CR	Conservatism Rate.
GI	Guaranteed Index.

List of Symbols

$A(t)$	The availability of the system at time *t*.
λ	Failure rate of the component.
μ	Repair rate of the component.
C_i	Component i.
[x]	Interval.
$\underline{x}$	Lower bound of the interval $[x]$.
$\overline{x}$	Upper bound of the interval $[x]$.
w([x])	Width of an interval $[x]$.
mid([x])	Midpoint of an interval $[x]$.
[A]	Interval matrix A.
$\underline{A}$	Lower bound of the interval $[A]$.
$\overline{A}$	Upper bound of the interval matrix $[A]$.
||*[A]*||	The norm of an interval matrix $[A]$.
w([A])	Width of an interval matrix $[A]$.
mid([A])	Midpoint of an interval matrix $[A]$.
n_f	The number of constraints f.
n_x	The number of variables x.
$\mathbb{X}_i$	The domain of a variable x_i.
X	The vector of variables.
[X]	The box of the vector *X*.
J_f	The Jacobian matrix of function f.

G	Stochastic graph.
V	Set of nodes.
U	Set of arcs.
$d^+(v)$	The external degree of the node v.
$d^-(v)$	The inner degree of the node v.
G_i	Set of performance level g_{ij} of component *i*.
g_{ij}	Performance level of component *i* where $j \in 1, 2, ..., K_j$.
$G(t)$	Performance level of the system at time *t*.
p_{ij}	The associated probability to the performance rate g_{ij} of element *i*.
$\boldsymbol{P}_i$	Set of the associated probabilities to the performance rates of element *i*.
$u_i(z)$	U-function of component *i*.
$P_i(t)$	Probability of the system to be in state *i* at *t*.
$W(t)$	Required demand.
X	Random variable.
$q_{ij}(t)$	Transition rate of the system from state *i* to state *j* at *t*.
S	The system.
E	Space of states.
e_i	The i^{th} state of the system.
Q	Transition matrix.
$P(t)$	Vector of probabilities of the system to be in a state.
π_i	Steady probability of the system to be in state *i*.
Π	Vector of steady probabilities of the system to be in a state.
$\mathbf{P}(0)$	Vector of initial probabilities of the system.
G_j	Component j.
p_i^j	The probability of being at state g_i^j.
$\lambda_{i,k}^j$	The transition or degrading rate from state g_i^j to g_{i-k}^j.
$\mu_{i,k}^j$	The repair rate from state g_i^j to g_{i+k}^j.

A_c	The availability of the system obtained by the technique of contractors.
A_s	The availability of the system.
A_e	The availability of the system obtained by the Exact method.
A_p	The availability of the system obtained by the Precise method.
C_s	The cost of the system.
f	The objective function.

Contents

Introduction

1. Context and Motivation

From the beginning, the human being seeks to find his own needs (food, sleep, comfort,...). However, what distinguishes the human of today with the one of the past is that over time he seeks to find these materials needs with the best benefits. In order to achieve his aim, he must use some "tools" that allow him to realize these needs. These "tools" are the products in the commerce. However, the choice of which product is chosen depends on its efficacy. In this context, the competition between companies has increased. When firms compete with each other, consumers get the best possible prices, quantity, and quality of goods and services. One important benefit of competition is a boost to innovation. Competition can lead companies to invent lower cost manufacturing processes, which can increase their profits and help them compete and then, pass those savings on to the consumer. That is why the evolution of technology has expanded which led to the increase of the complexity of systems (products) and has further reduced their design and manufacturing costs. Correspondingly, manufacturers rely on the criterion of quality to be distinguished in the commerce/industry. To achieve this goal, they must control the various tools that will enable them to keep in a competitive position and must take actions of improvement at all levels. All these reasons make that **Dependability** domain has an undeniable means that must be mastered when designing any system.

Dependability has become a necessity in the industrial world during the twentieth century. The term "dependability" appeared in an advertisement on Dodge Brothers engines in the 1930s. The aim of dependability is to achieve the "grail" of system design. In order to realize this aim, all possible uses of a product should be tested over a long period, which is unthinkable in the industrial context or even impossible to achieve in any case. Dependability is an activity domain that proposes means to increase the attributes of the system in a reasonable time and with a less cost.

Dependability is often referred to as "the science of failures"; it includes their knowledge, evaluation, foresight, measurement and mastery. A crosscutting area requires a global knowledge of the system such as the conditions of use, the external risks, the functional and material architectures, the structure and fatigue of the materials. In systems engineering, dependability is defined as the property that enables system users to place a justified confidence in the service it delivers to them and it is a measure of a system's availability, reliability, and its maintainability, and maintenance support performance, and, in some cases, other characteristics such as durability, safety and security [130]. The key concept that our work is based on is the **availability**. The availability $A(t)$ is the ability of a system to be operational at a specific moment [130].

The cost of a high level of quality is very expensive. The designer must compromise between the availability and the economic costs. Users can reject systems that are unsafe, unreliable or insecure. Thus, to have a good system in terms of high availability, the cost of system can be extremely high. Therefore, any user (or industry) will ask the following question before getting any product: "What is the optimal product in the market?" To answer this question, we must combine the following two points:

- The best availability of the system: The user wants a product that lasts as long as possible.
- The best cost of the system: The user wants a product with moderate price.

2. Problem statement

Since, dependability domain is extremely large and complex, in this thesis we focus on the system's availability assessment. Availability is an important parameter to study the system's working states. It consists in that a component, a repairable unit, a replaceable unit or a system is in a state of normal operation at a given instant or during a given period. Availability calculation is based primarily on knowledge of failure rates and repairs of system components [118]. Availability analysis helps to calculate the ability of a system to provide a required level of performance depending on the level of degradation.

Several methods have been used to calculate the availability of a system, among them we find the Universal Generating Function (UGF) [63], Inclusion-Exclusion technique [77, 76], Markov models [96],... These methods employ different probabilistic techniques to evaluate this criterion, but these proposed approaches remain effective only for very specific

cases, for example the cases of binary systems.

A binary system is a system where only two cases are possible: perfect functioning and total failure. While the transition to multi-state systems drastically restricts the application of most of these methods. In real life, the systems correspond to multi-states systems. In such scenarios, systems and their components can operate at different performance levels between working and failure states. However, the evaluation of the availability of the multi-state systems is more difficult than in the binary case, because we have to take into account the different combinations of the component failure modes. Throughout this thesis, we search for a method that helps us to compute the availability of multi-states systems.

In addition to the multi-state aspect, there is the structural aspect of the system, which illustrates the type of connection connecting the components. In this case, the system become much complex. In general, most methods of availability assessment can be applied efficiently only to systems with a simple structure: series, parallel, bridge or mixed,. . .

In this thesis, we are interested in the state of the system after a long period of time, which means that we need to calculate the steady availability of the system. Therefore, we use Markov models to represent the complex multi-states system and to calculate its availability.

On the other hand, in dependability studies, the knowledge that we have about the components availability is generally imprecise. Taking into account totally or partially of the imprecision of the used knowledge involves on the validity of the results. This requires some methods that allow modeling and manipulation of such imprecision as the probability theory, the fuzzy set theory, the belief function theory,. . . However, in our work we want to calculate the imprecise availability using Markov models. For our best of knowledge, there is only one work in the literature that aims to solve this problem [110].

The first part of our work consists on proposing a methodology to calculate the imprecise availability of a multi-states system. The second part consist on proposing a methodology that optimize the imprecise availability, in other words, propose a methodology that chooses the best system with the best availability and the best cost?

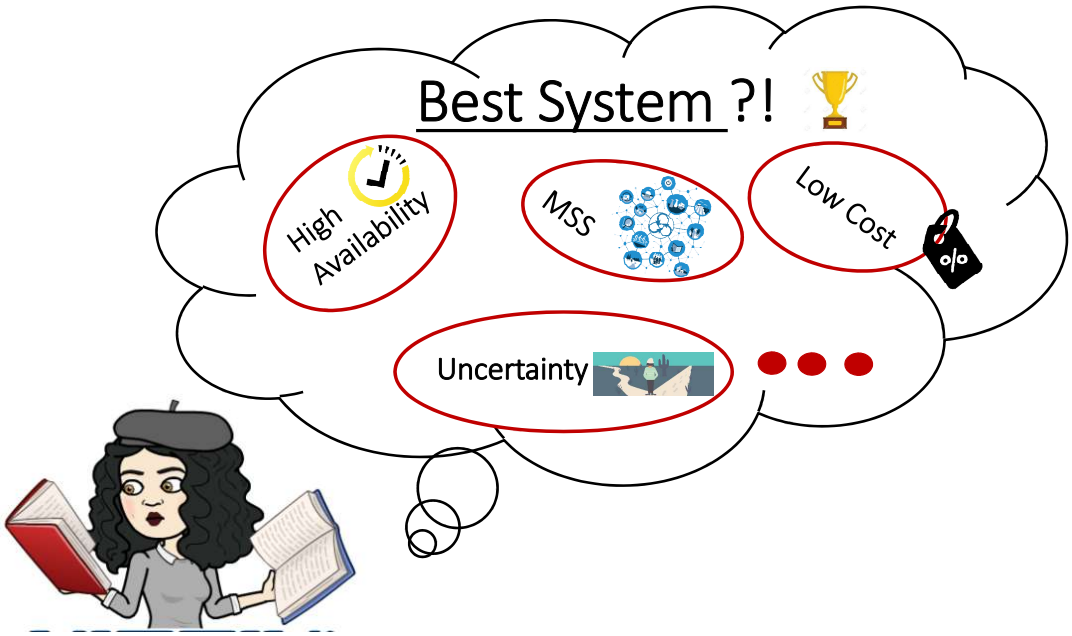


Figure 1: Representation of the thesis problematic

3. Objectives and Contributions

In this thesis, we focus on availability analysis of the multi-states systems. In this context, our aim is to propose a methodology that optimize the design of multi-states systems. We will rely on the cost criterion on one side, and the availability criterion on the other side, in order to find the best compromise between these two criteria. Therefore, two objectives must be done:

- First objective: Proposition of a method that compute the imprecise availability of multi-states systems.
- Second objective: Proposition of a method that optimize the imprecise availability of multi-states systems.

First, we propose to use the Markov models [37] to model the multi-states systems and to calculate the steady imprecise availability of the system. However, in our study, we must take into account the uncertainties on the data of the multi-states system. In our work, we propose to use interval analysis [89, 23] to solve the problem of uncertainties and to treat the data with uncertainties in terms of intervals. To compute the imprecise availability, we

introduce for the first time in dependability domain, a technique used in interval analysis, the "Technique of Contractors" [10, 49]. We apply this technique to large systems.

After we define the principal methodology of computing the imprecise availability of multi-states systems, we pass to the next step. We need to propose a method that aims at optimizing the imprecise availability of a multi-states system, by choosing a certain number of components among different components that provide the best availability with the lowest cost for some system's configuration. As a simple start, we proceed in two ways. The first consists in choosing among several components with different performances, the best components of the system, for a given configuration. The second is to define a fixed number of possible components, and choose the best configuration of the system that provides the best availability and the best cost.

4. Thesis outline

This manuscript comprises two parts: the first lumps prerequisites and background of the thesis while the second presents our main contributions. The hierarchy of the report is based on five chapters as indicated in what follows.

The first part is devoted to present basic knowledge related to our subject, the basic notions and the state of the art.

Chapter 1 provides the scientific context for our work. We first start with an overview on dependability, then we present a brief history of dependability. Since the subject of the thesis is based on uncertainties, we give a short explanation of aleatory uncertainty and epistemic uncertainty. Next, we define the main concepts in dependability (RAMS for Reliability, Availability, Maintainability and Safety). Then, we continue with a presentation of the RAMS assessment for the component that belongs to a system. After, we discuss some of the RAMS methods that help to illustrate the system and to calculate RAMS. To summarize the chapter, we end with a conclusion.

Since uncertainty are, mainly modeled in intervals in our work, chapter 2 is devoted to present the basics of interval analysis.
First, we start with a historical overview of interval analysis. Second, we explain the main concept of interval analysis. Third, we focus on giving an overview of the basic terms and definitions. To the good use of intervals, we present later the interval arithmetic computa-

tion principles and we give the well-known properties of intervals (commutativity and associativity, additive and multiplicative identity elements, nonexistence of inverse elements, sub-distributivity, cancellation). The main use of intervals in the thesis appears in matrices, therefore a section is dedicated to present the concepts of interval matrices. We present also a well-known technique in interval analysis which is "constraint satisfaction problem". In this section, we give a brief history of contractors, the main definitions, some of the contractors, and we focus on the forward-backward contractor since it will be applied in our methodology. Finally, we end this chapter by a conclusion.

Chapter 3 is devoted to present the related work to imprecise availability assessment of multi-states systems. We start with a distinction between binary systems and multi-states systems. Then, we present the techniques used to calculate the availability of a binary system. After, we explain the case of multi-states system and we present some of the methods (UGF, Inclusion-Exclusion technique, Monte-Carlo simulation, Markovian method) used to calculate the precise availability of such systems and their examples in the literature. Ee present in this chapter the imprecise case of availability assessment and we cite some of related works. We end the chapter with the positioning of our work compared to others and with the conclusion of the chapter.

Part 2 is organized into two chapters, that consist of the contributions of this thesis.

Chapter 4 provides the first contribution of this thesis, which is a method of computing the availability of multi-states systems using imprecise Markov models. First, we start with some basics of the use of imprecise Markov models in dependability and we discuss related works. To compute the imprecise availability of MSS using Markov models, we present two methods. The first called the "Exact method" and the second, which constitute one of the main contributions of this thesis, is based on "the technique of contractors". We give the metrics for validation of the methodology and provide numerical examples in order to assume better understanding of the new proposed approach. We end up with the discussion of the methods and conclusion.

Chapter 5 provides the second contribution of the thesis, which is the optimization of the imprecise availability of multi-states systems using the method proposed in Chapter 4. We start with an introduction about the optimization in dependability in general. After, we present two of the possible cases of optimization of the imprecise availability and provide useful and clear solutions. Two examples are presented to illustrate the two methodologies.

We end with the results discussion and with the conclusion of the chapter.

A general conclusion recapitulating the basic concepts and contributions of the thesis, as well as future work and perspectives, are given at last.

Part I

Background

Chapter 1

Dependability basics

1.1 Introduction

Nowadays, dependability domain is one of the major challenges that interest the manufacturers and operators in the various sectors (automotive, aeronautics, aerospace, telecommunications, nuclear, defense, etc.). The exponential evolution of technology has increased the complexity of the systems and further reduced their design and maintenance costs. In this perspective, manufacturers rely on quality criteria to be distinguished in the market. To do this, they must master the various tools that enable them to make it possible to keep their competitive position and must adopt improvement actions to all levels. For all these reasons, dependability is the key to success which must be mastered when designing any system.

Dependability provides users measures to evaluate services supplied by systems. It is widely used in many domains, including human factors [107], railway systems [92], nuclear systems [122, 55], etc.

In this chapter, we are interested in different concepts related to systems dependability. At first, we recall the definition of a system and we focus on the terms of dependability by exposing its hindrances, its aspects and a brief history in order to become familiar with the subject. So, we first list the precise definitions characterizing the various concepts in dependability which are given by Avizienis *et al.* [48], and then recall other existing definitions. After, we present the different mathematical measures and theories of the reliability and availability of the systems that discern these concepts. In addition, we highlight the useful methods of analyzing the reliability/availability through a bibliographic study. Similarly, we present the different structures of existing systems. Finally, we show the difference between binary and multi-state systems (MSS) and the basic techniques that help to calculate

the reliability/availability.

A **system** is an entity with bounds (frontiers) which is able to have autonomous activities according to objectives and interacting with other entities. The **function** of a system is what the system is intended to do and is described by the functional specification in terms of functionality and performance. The **behavior** of a system is what the system does to implement its function and is described by a sequence of states. The **total state** of a given system is the set of the following states: computation, communication, stored information, interconnection, and physical condition. The **structure** of a system is what enables it to generate its functional organization. The **service** delivered by a system is its behavior as it is perceived by its users.

1.2 What is the concept of dependability?

The concept of dependability is defined by many, in this section we cite the definitions introduced in some references.

This concept is defined according to Mortureux [83] as a set of means and a set of results produced by these means. For a system, this notion is considered as the characteristic of a system that allows to place a justified confidence in it. This confidence is based on a set of steps and is expressed through a set of characteristics, in particular availability.

According to Villemeur [115], dependability is considered as the ability of an entity to perform one or more required functions under given conditions. In this definition, the entity can refer to an organization, system, product or means and the function of the system means the functional performance expected by the system. He also notes that this concept mainly encompasses reliability, availability, maintainability and safety, but also other skills such as durability, testability. . . or combinations of these skills. In a broad sense, this concept refers to the science of failures and breakdowns.

Laprie [62] says that dependability is the property that allows users of the system to place justified confidence in the service it provides them. Thus, according to this definition, dependability reflects the trust that can be placed in a system.

1.3 Dependability history

Dependability problems [4] have existed for a very long time, as soon as a system may have failed.

The first collection of statistical information on engines and aircraft accidents was conducted in the 1930s, in the air transport sector. Between 1939 and 1942, the very first quantified objectives were given by Captain A.F. Pugsley of the 7th Canadian Infantry Brigade appeared. He estimated failure rates of $10^{-5}/h$ for aircraft and $10^{-7}/h$ for their structures.

Between 1940 and 1950, the concept of maintainability appeared and a discipline started developing under the name of "reliability theory", following a comparison of the failure frequencies of aircraft used during the Second World War. It was applied to electronics in the aeronautics, defense and nuclear sectors.

Between 1960 and 1970, the generalization of this approach to other components: mechanical, hydraulic, electrical, then to people, software ... and the development of new methods (Process Hazard Analysis (PHA), Fault Trees (FT),...) to control the risks, have been done.

In 1962, the Academy of Sciences welcomed the word "reliability" in its terminology.

In the 1980s, a formalization of the global approach of the dependability on the concept of complex systems and the appearance of several deepening were manifested in the development: reliability databases, modeling and analysis methods, calculation software, modeling software, etc, have been defined [62].

To conclude this brief history of dependability, we can say that today the dependability of a system has never been more important. The study of the dependability of any system is well needed to be recognized in the industry in all kind of majors.

1.4 What are uncertainties?

In the work of Aven [5], uncertainties are usually divided into two types: **aleatory uncertainty** and **epistemic uncertainty**. Aleatory uncertainty is due to the natural variability of a random phenomena. It reflects the objective feature of the world. Aleatory uncertainty is usually represented by the probability models and frequentist probabilities. Epistemic

uncertainty expresses the lack of knowledge about the true values of the frequentist probabilities and parameters of probability models. It reflects the subjective feature of the analyst. The distinction is important because epistemic uncertainties can be reduced by acquiring knowledge on the studied system, whereas aleatory uncertainties cannot. Furthermore, some works have proven that uncertainties in reliability and risk assessments are mainly epistemic [121].

Due to the environment and the lack of failure data (due to rare failure events) related to some components or subsystems used in transportation systems, there are epistemic uncertainties when modeling such systems. Because of this, there is always uncertainty in the modeling of transportation systems. As transportation systems are high-availability systems, the influence brought by epistemic uncertainty cannot be ignored. This motivates us to consider all types of uncertainties in our models. Keep in mind that there are other points of view as to how to distinguish sources or types of uncertainty [31]. Blockley [15] argued that uncertainty is classified using three conceptually distinctive and orthogonal attributes: fuzziness, incompleteness (epistemic) and randomness (aleatory). The motivation was that the aleatory/epistemic classification is not rich enough for practical decision making. Blockley further described some characterizations of uncertainty, such as ambiguity, confusion, contingency, indeterminacy and conflict emerge from mixes and interactions between these basic three attributes. As a consequence, several uncertainty theories for uncertainty quantification were presented, including Bayesian theory, imprecise probability theory [117], possibility theory [53], belief functions theory [26], etc. Probability is the predominant tool used to measure uncertainties in reliability and risk analysis. However, some researchers have argued that probabilistic approaches are not adequate to handle epistemic uncertainties in reliability studies, and should be restricted to aleatory uncertainties [5]. The Bayesian approach requires us to specify probability distribution. But, in many cases, prior knowledge is either vague, or non-existent. Thus, some recent theories such as belief functions were introduced to handle both aleatory and epistemic uncertainties in reliability studies. During the last years, belief functions theory was applied in a few works concerning reliability and risk analysis [41, 47, 98]. Particularly, a proper reliability model using belief functions theory was introduced by Sallak *et al.* [98] and further studied in [101, 99]. In these works, different points were treated concerning the representation of the system's configuration, the failure dependencies between components and a comparison with other uncertainty methods.

1.5 Concepts in RAMS assessment

The system is characterized by five fundamental properties: functionality, usability, performance, cost, and dependability. The concept of dependability as we consider in this thesis denotes the ability to deliver service that can justifiably be trusted [91].

A systematic exposition of the concepts of dependability consists of three parts: the threats to, the attributes of, and the means by which dependability is attained, as shown in Figure 1.2.
RAMS is an acronym for Reliability, Availability, Maintainability, and Safety, it denotes the concept of dependability. Dependability is an integrating concept which contains the following attributes [6]:

- **availability**: readiness for correct service.
- **reliability**: continuity of correct service.
- **safety**:
    1. **technical security:** absence of catastrophic consequences on the users and the environment.
    2. **regulation security:** with respect to human behavior (compliance with the labor code, anti-intrusion resistance to malice).
- **integrity**: absence of improper system alterations.
- **maintainability**: ability to undergo modifications and repairs.
- **Security** is defined as the combination of **confidentiality**, the prevention of the unauthorized disclosure of information, **integrity**, the prevention of the unauthorized amendment or deletion of information also it takes into account protection against physical "attacks" (destruction, sabotage,. . .), and **availability**, the prevention of the unauthorized withholding of information [22, 90].

There are some threats to dependability: failures, errors, and faults. A **failure** occurs when the service delivered by the system deviates from what is required. The cause of a failure is an **error** affecting a part of the system's state (e.g. erroneous value of a variable). The cause of an error is a **fault** (e.g. short-circuit on an electronic component). Note that these definitions are in fact recursive because a failure of a component is a fault for the system containing that component. The causal chain is presented in Figure 1.1.

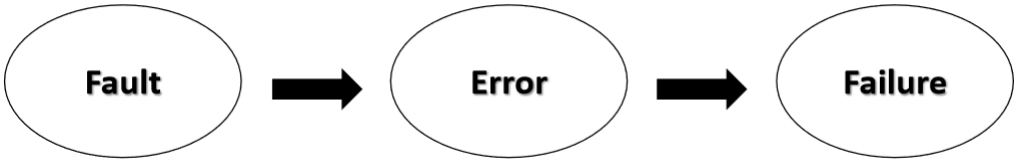


Figure 1.1: The chain between the threats of dependability [6].

Faults during the life of a system are classified into three groups [6]:

- **development faults** that include all fault classes occurring during development,
- **physical faults** that include all fault classes that affect hardware,
- **interaction faults** that include all external faults.

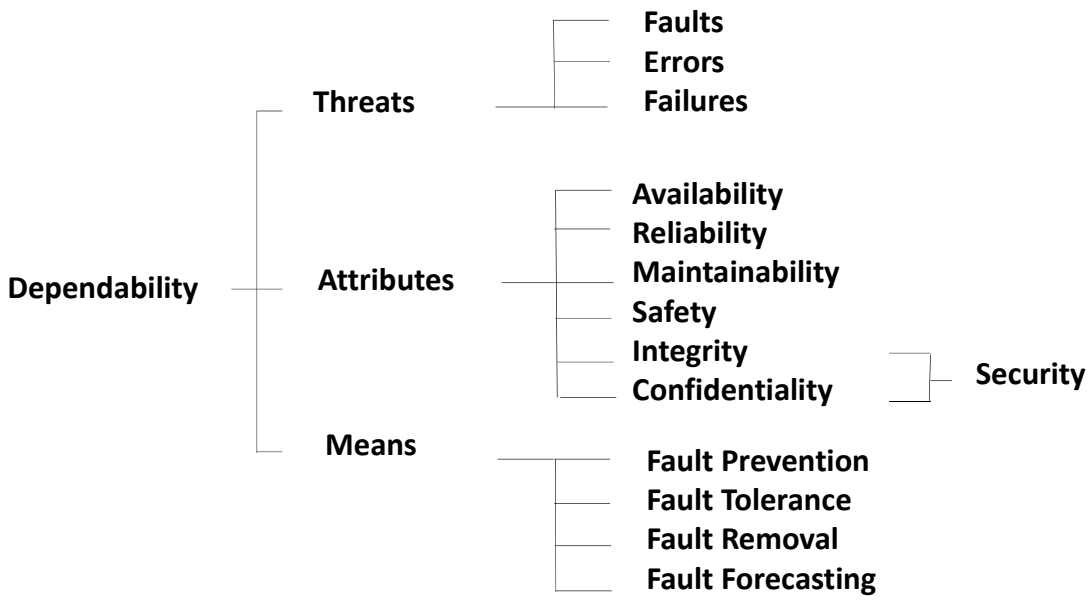


Figure 1.2: The dependability tree [6].

A system does not always fail in the same way. The ways in which a system can fail are its failure modes. These can be ranked according to failure severity. The modes characterize incorrect service according to several means, which they are developed to attain the various attributes of dependability. Those means are grouped into four major categories:

- **Fault prevention** means to prevent the occurrence of introduction of faults.

- **Fault tolerance** means to avoid service failures in the presence of faults.
- **Fault removal** means to reduce the number and severity of faults.
- **Fault forecasting** means to estimate the present number, the future incidence, and the likely consequences of faults.

1.6 RAMS assessment at the component level

RAMS (Reliability, Availability, Maintainability, Safety) attributes are always used to assess the performance of a system [14]. In this section, some quantitative measures of RAMS assessment are applied at the component level.

1.6.1 Reliability

Reliability is defined as the ability of a good to perform a required function under given conditions for a given period of time according to the standard (NF EN 13306, French Standard (European Norm), Maintenance Vocabulary, Maintenance Terminology, October 2001).

Reliability is the support to be given to increase availability while taking into account the objective of cost optimization. This quantity can be quantified by these two indicators: the mean time to failure (MTTF) for non-repairable systems, such as lamps and electronic components, and the mean time between consecutive failures (MTBF) for repairable items, such as industrial or domestic equipment. It represents the probability $R(t)$ that entity E performing its functions at the instant 0 always accomplishes them at the instant t. It is characterized by its function $R(t)$ and its failure rate $\lambda(t)$.

$$R(t) = Probability(No\ failure\ on\ the\ interval\ [0,t]) \tag{1.1}$$

where the component or the system is supposed to be working at time t=0.

In this context, the unreliability function is also introduced, denoted as F(t). Unlike reliability, this function represents the probability that entity E has experienced a failure before time t.

$$F(t) = Probability(failure\ on\ the\ interval\ [0,t]) \tag{1.2}$$

with $F(t) = 1 - R(t)$.

Note that the reliability function, can take forms beyond time depending on the context of the work. For instance, it may refer to the number of cycles carried out for a valve or contractor, at the distance traveled for a car, at the number of revolutions for a pump or motor, etc.

Generally, to measure reliability, the function of the failure rate is used. The failure rate λ represents the frequency at which a component or a system fails. The failure rate at time t is defined as

$$\lambda(t) = \frac{P(Failure\ on\ [t, t+dt]\ with\ no\ failure\ on\ [0,t])}{P(no\ failure\ on\ [0,t])} \tag{1.3}$$

$$\lambda(t) = -\frac{1}{R(t)}\frac{dR(t)}{dt} \tag{1.4}$$

In case the failure rate is constant and thus does not depend on time, the reliability function follows an exponential law:

$$R(t) = e^{-\lambda t} \tag{1.5}$$

1.6.2 Availability

According to the standard (NF EN 13306), availability is defined as the ability of a component or a system to perform a required function under given conditions at time t, assuming that the provision of the necessary external means is ensured. The availability at time t is expressed by

$$A(t) = Probability(System\ is\ working\ at\ time\ t). \tag{1.6}$$

In this thesis, we are interested in availability calculation.

For non-repairable items, the notion of availability remains the same as that of reliability. As long as for repairable items, this concept is more valid when breakdowns are rare and short, i.e. when the item is reliable and maintainable.

Availability is often quantified by these two indicators: Mean Time to Failure (MTTF) and Mean Time to Repair (MTTR). Then the average availability A_i can be evaluated by the ratio:

$$A_i = \frac{MTTF}{MTTF + MTTR} \tag{1.7}$$

Mean Time To Failure (MTTF) is the average operating time before the first failure. When t tends to infinity, $R(t)$ decreases (as fast as an exponential law), we can calculate the MTTF as follows:

$$MTTF = \int_0^\infty R(t)dt \tag{1.8}$$

For a component with a constant failure rate λ, we have

$$MTTF = \frac{1}{\lambda} \tag{1.9}$$

Mean Time To Repair (MTTR) is the average time of repair, it is defined in 1.6.3.

1.6.3 Maintainability

The standard ($NFEN13306$) defines maintainability as the ability of a component or a system to be maintained or repaired during a given time interval, when maintenance is performed under given conditions, with prescribed procedures and means. It is characterized by the probability $M(t)$ that at time t an entity E is in a state performing these functions knowing that it was in failure at time 0. The maintainability at time t is expressed by:

$$M(t) = Probability(Repair\ completed\ on\ [0, t]) \tag{1.10}$$

where the component or the system is supposed to be failed at time t=0.

From its definition, it is clear that this concept can only be applied to repairable systems, where the determination of maintenance policies is very important. On the other hand, for non-repairable systems, the notion of maintainability does not apply. Generally, this quantity is measured by the following two indicators, the average repair time (MTTR) and the repair rate μ.

The repair rate μ represents the frequency at which the failure of a component or a system is repaired. For a repairable component or system, the repair rate at time t is defined as:

$$\mu(t) = \frac{1}{1 - M(t)} \frac{dM(t)}{dt}. \tag{1.11}$$

For a component with a constant repair rate μ, the maintainability at time t is given by:

$$M(t) = 1 - e^{-\mu t} \tag{1.12}$$

Mean Time To Repair (MTTR) is the average time of repair. It can be calculated as follows, such that $M(t)$ tends to 1 (as fast as an exponential law), when t tends to infinity:

$$MTTR = \int_0^\infty (1 - M(t))dt \tag{1.13}$$

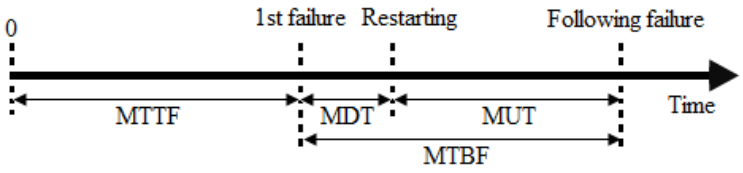


Figure 1.3: Illustration of MTTF, MUT, MDT, MTBF.

For a component with a constant repair rate μ, we have

$$MTTR = \frac{1}{\mu} \tag{1.14}$$

For a component with constant failure rate λ and constant repair rate μ, the availability is defined as

$$A(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}e^{-(\lambda+\mu)t} \qquad (Component\ is\ working\ at\ t=0) \tag{1.15}$$

$$A(t) = \frac{\mu}{\lambda+\mu} - \frac{\mu}{\lambda+\mu}e^{-(\lambda+\mu)t} \qquad (Component\ is\ failed\ at\ t=0) \tag{1.16}$$

Mean Up Time (MUT) is the average operating time after the repair. Mean Down Time (MDT) is the the average time of repair after a failure. Mean Time Between Failures (MTBF) is the average time between failures. Figure 1.3 illustrates all these parameters. According to this figure, we have the following equations:

$$MTBF = MDT + MUT \tag{1.17}$$

$$Availability = \frac{MDT}{MTBF} \tag{1.18}$$

1.7 RAMS Methods

In the previous section we introduced the RAMS concepts, in this section, we talk about the RAMS methods.

Dependability combines a set of techniques used to identify, analyze, manage and, reduce the risks associated with industrial systems [86]. All the dependability methods have at least three points in common, which can be summarized in three types of action:

- Identify the processes that can affect reliability, maintainability, availability or security.
- Model these different processes to facilitate the understanding of the involved mechanisms.

- Evaluate the analysis results by using the obtained models to assess the level of dependability of the studied system.

Many works offer an inventory of these methods, among them Amongst other presentations [116, 11, 87, 133, 86].

Three classification modes are commonly used:

1. **Qualitative/quantitative approaches**

   - *Qualitative/Quantitative Approaches*
     The results provide information about system characteristics: weak points of the system, influence of a given element on the reliability of the system, identification of critical paths, testing (for critical paths) of disposal methods.
   - *Quantitative approaches*
     The results are due to the calculation of reliability, availability,. . .

2. **Inductive/deductive approaches**

   - *Inductive approaches*
     Inductive approaches based on a bottom-up approach, they consider an initiating event (technical failure, organizational dysfunction) which they seek to characterize the consequences on the system and its environment.
   - *Deductive approaches*
     Deductive approaches based on a top-down approach, they consider a dreaded event (system shutdown, malfunction) which they seek to explain the causes, usually in the form of sequences of events.

3. **Static/dynamic approaches**

   - *Static approaches*
     Static approaches allow to analyze the system from a structural point of view without taking into account changes over time; generally they rely on a Boolean mathematical model of the system that will provide, for example, combinations of failures leading to the fail of the system but without representing the temporal interrelations that affect it.
   - *Dynamic approaches*
     Dynamic methods take into account the behavioral and temporal aspects.

A predictive safety analysis is a way to study a real system in order to produce a model of the system related to dependability characteristic (reliability, availability, maintainability, security). The elements of this model will be events likely to occur in the system and its environment, such as:

1. Failures of system components.
2. Environmental events.
3. Human errors in operation phase.

Thus, thee model represents all the failures of system components that compromise one of dependability characteristics. Several methods of analysis have been developed. The main ones are:

- Preliminary Hazard Analysis (PHA).
- Failure Modes and Effects Analysis (FMEA).
- Reliability Block Diagram (RBD).
- Truth-Table method (TTM).
- Fault Tree Analysis (FTA).
- Accident Consequence Tree (ACT).
- Cause-Consequence Diagram Method (CCD)

In this section, we present some of the classical methods of RAMS that corresponds to the structural aspect of the systems in the aim of availability computing.

1.7.1 Graphical representation of a system

Any system is designed to perform a specific function. The system is defined by the relationships that these components have with each other. According to Kjarulff and Madsen [57], a system able to perform tasks that are supposed to be intellectually demanding will often exhibit artificial intelligence. It is said *expert system* if the system's problem solving ability is restricted to a particular area of expertise.

Graphical models are usually used to visualize the structure of a system, the behavior of a system, and the conditional relationship between components. A graphical model can be constructed to solve a particular problem within a given problem domain, thus expert system

can refer to systems that perform reasoning and decision making by means of graphical models. In the literature, there are some widely used graphical models. In this section, some useful graphical models are presented. Users choose a graphical model according to their objectives and the characteristics of these graphical models.

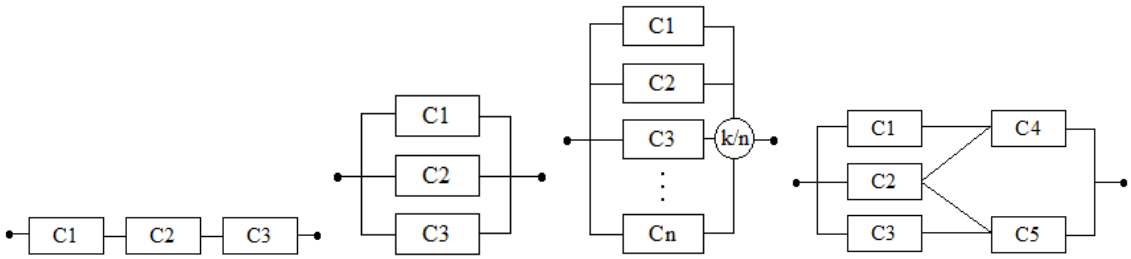


Figure 1.4: RBDs for some typical systems (series system, parallel system, k-out-of-n system, complex system [91])

1.7.1.1. Reliability block diagram

For the development of any system, it is not only necessary to identify its components but it is also important to know the layout and connection that will bring these elements together. Following the different interactions between components, the system is oriented towards achieving an expected objective. The term system is used in all fields: mechanical, automatic, electronic, computer, etc.

Reliability block diagram (RBD) is the most common model that has been used in reliability analysis for many years. A number of components represented by blocks are connected to reflect the logical reliability structure of a system. In this section, we expose the different system configurations, as illustrated in Figure 1.4.

A- Systems with simple configuration

The main configurations often encountered in the context of dependability analysis are exposed hereafter:

Series system

A series system is characterized by a linear sequence of n elements (cf. Figure 1.5). According to this structure, the failure of one of the n components leads to the failure of the whole system, which means that the system is working when all components are working.

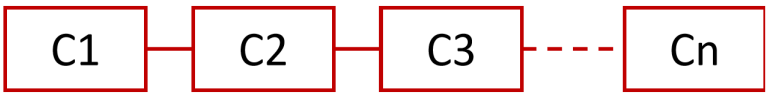


Figure 1.5: Series system

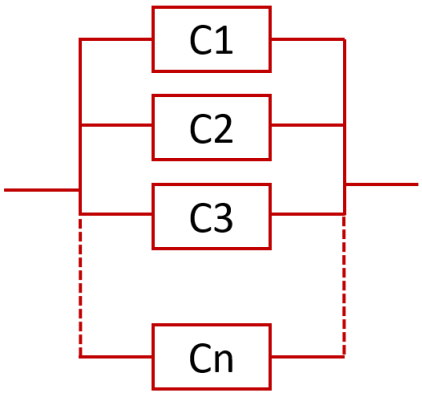


Figure 1.6: Parallel system

The reliability of the whole system R_s is equal to the product of the reliability of each component:

$$R_s = \prod_{i=1}^{n} R_i \tag{1.19}$$

Parallel system

A parallel system is characterized by a parallel association of all components (cf. Figure 1.6). Generally, the failure of one or more elements does not cause the system to fail, the system only fails if all the elements fail.

The failure probability of the system F_s is equal to the product of the failure probability of each component:

$$F_s = \prod_{i=1}^{n} F_i = \prod_{i=1}^{n} (1 - R_i) \tag{1.20}$$

So, the reliability R_s of the system is:

$$R_s = 1 - F_s = 1 - \prod_{i=1}^{n} (1 - R_i) \tag{1.21}$$

Series-parallel system

The series-parallel system consists of n subsystems connected in parallel such that each subsystem consists of k elements placed in series (cf. Figure 1.7). A series-parallel system

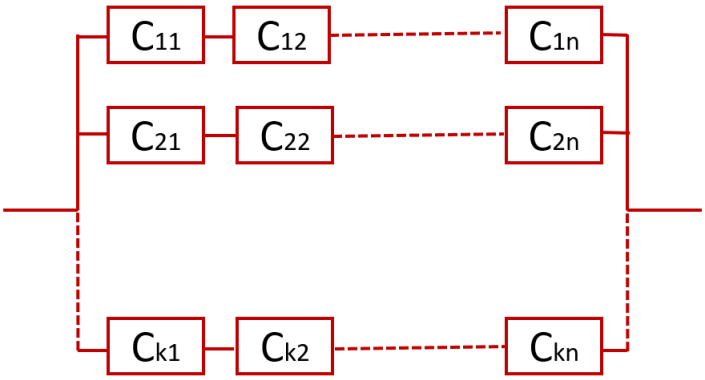


Figure 1.7: Series-parallel system

is the result of the combination of both series and parallel systems.

To calculate its reliability, the complete system is reduced to a parallel system by modeling each subsystem in series with a single component. The reliability of a series subsystem i is given by:

$$R_i = \prod_{j=1}^{n} R_{ij} \tag{1.22}$$

So, the reliability R_s of the complete system is:

$$\begin{aligned} R_s &= 1 - \prod_{i=1}^{k}(1 - R_i) \\ R_s &= 1 - \prod_{i=1}^{k}(1 - \prod_{j=1}^{n} R_{ij}) \end{aligned} \tag{1.23}$$

Parallel-series system

The parallel-series system consists of n subsystems connected in series such that each subsystem consists of k elements placed in parallel (cf. Figure 1.8). Similarly, a parallel-series system is the result of combining the two series and parallel systems.

To calculate its reliability, the complete system is reduced to a series system by modeling each subsystem in parallel with a single component. The reliability of a parallel subsystem j is:

$$R_j = 1 - \prod_{i=1}^{n}(1 - R_{ij}) \tag{1.24}$$

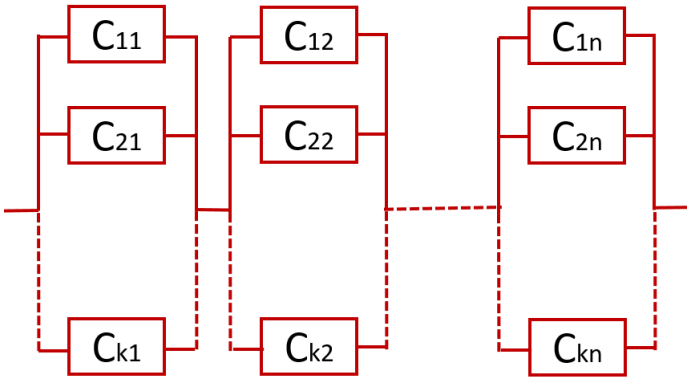


Figure 1.8: Parallel-series system

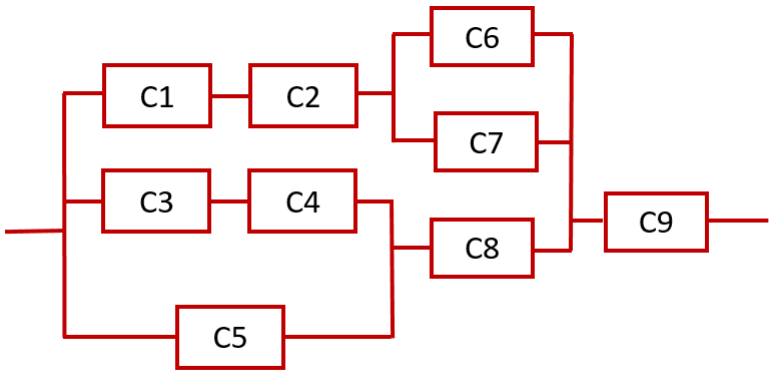


Figure 1.9: Mixed system

So, the reliability of the whole system R_s is as follows:

$$\begin{cases} R_s = \prod_{j=1}^{n} R_j \\ R_s = \prod_{j=1}^{n} (1 - \prod_{i=1}^{k} (1 - R_{ij})) \end{cases} \tag{1.25}$$

Mixed system

A mixed system is the combination of series and parallel structures (cf. Figure 1.9). The reliability of the complete system is assessed by decomposing the system into several series and parallel subsystems, then each subsystem is reduced to a single component.

B- Systems with any kind of configuration

Other types of system configurations are presented hereafter:

k-out-of-n system (k/n)

A redundant system k-out-of-n only works if at least k components of the n parallel

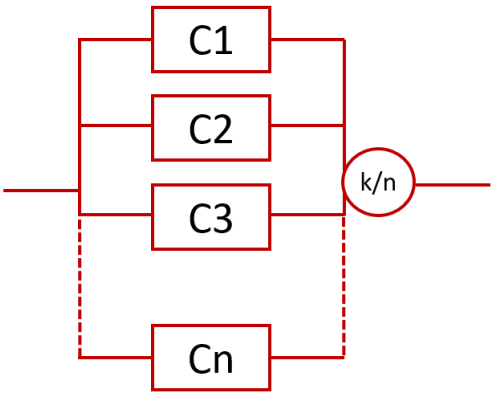


Figure 1.10: k-out-of-n system

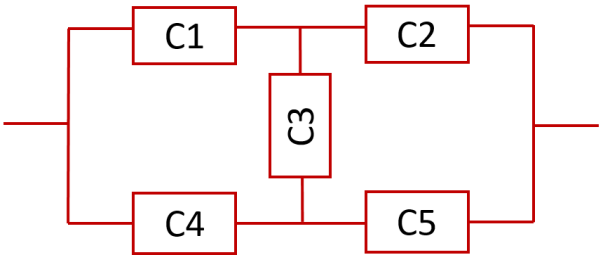


Figure 1.11: Bridge system

components work to guarantee the system function, Figure 1.10. In the case where all the components of the system are identical, i.e. they have the same failure rate and reliability R, the system's reliability R_s is equal to the sum of the probabilities of all configurations with at least k operational components:

$$R_s(k,n) = \sum_{i=k}^{n} C_n^i R^i (1-R)^{n-i} \; ; (C_n^i = \frac{n!}{i!(n-i)!}) \tag{1.26}$$

Bridge structure

A system is said to have a bridge structure when the system cannot be broken down into series and parallel combinations (cf. Figure 1.11). This system operates in parallel-series mode under the control of the bridge component C_3. If this component fails, the system switches to series-parallel mode, which is considered degraded mode.

To calculate the reliability of the system R_s, either the Boolean table is used while listing all the possible combinations of component states, either iteration system is reduced using

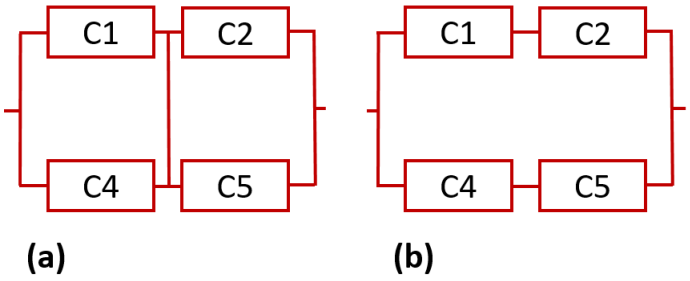


Figure 1.12: Decomposition of a bridge system

the conditional probability theorem.

Figure 1.12(a) represents the case where the bridge component is working. The reliability is given by:

$$R_a = [1 - (1 - R_1)(1 - R_4)].[1 - (1 - R_2)(1 - R_5)] \tag{1.27}$$

Figure 1.12(b) represents the case when the bridge component fails. we have:

$$R_b = 1 - (1 - R_1R_2)(1 - R_4R_5) \tag{1.28}$$

So, the reliability R_s of the complete system is:

$$R_s = R_3.R_a + (1 - R_3).R_b \tag{1.29}$$

Systems with complex configurations

A system with a complex configuration can only be represented by its connection diagram (cf. Figure 1.13). In this example, it is possible to represent it by "bridge" or simple structure, however it must be combined (or with conditional probabilities).

To go from the starting point (S) to the arrival point (T), it is necessary to pass through a set of linearly placed subsystems, each subsystem consists of a set of components and each component can be connected with one or more components of the previous and the next subsystems.

Indeed, the other types of configurations represent particular cases of the complex system, for example, when it is a parallel system, any component of the system is connected to all the components of its two preceding and following subsystems. The mastery of this type of configuration will allow us to master all the other configurations that follow.

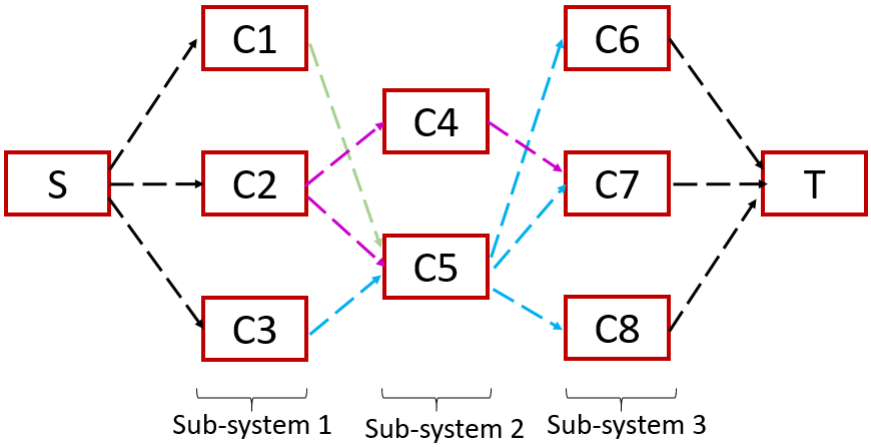


Figure 1.13: Complex system

In the example of Figure 1.13, we can apply some approaches to calculate the reliability/availability of such system. Although the complex system represents the most general case of systems that may exist, however, there is more complicated cases then the one in Figure 1.13, for example when the system involves evolution over time and/or dependencies (example of performance distribution or non-independent failure between the components, etc.).

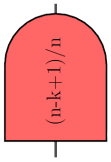
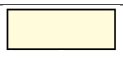
graphic symbol	meaning
	or gate
	and gate
	(n-k+1)-out-of-n gate
	top or intermediate event
	elementary basic event

Table 1.1: Graphical symbols in FT.

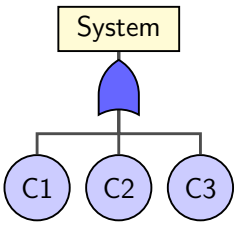


Figure 1.14: Series system FT

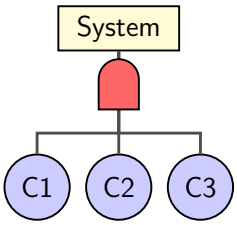


Figure 1.15: Parallel system FT

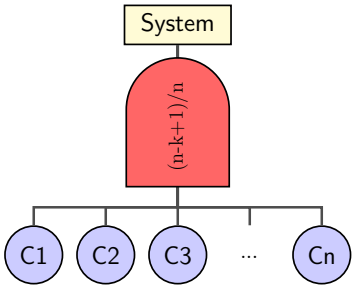


Figure 1.16: k-out-of-n system FT

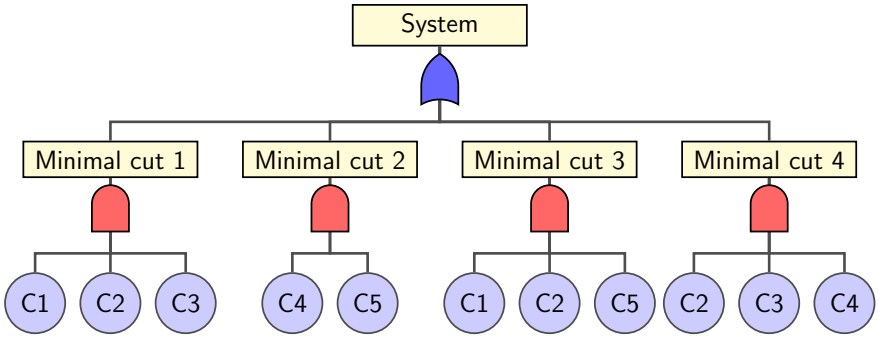


Figure 1.17: Complex system FT

Figure 1.18: FTs for some typical systems in Figure 1.4.

1.7.1.2. Fault tree

The Fault Tree Analysis (FTA) is an analytical technique that is used for Reliability, Maintainability and Safety Analysis [58]. It is a top-down (deductive) analysis, proceeding through successively more detailed (i.e. lower) levels of the design until the probability of occurrence of the top event (the feared event) can be predicted in the context of its environment and operation.

The FTA was used for the first time in 1962 [21, 50], for the US Air Force by Bell Telephone Laboratories on the Minuteman Weapon System [33].

FTA received extensive coverage at a 1965 System Safety Symposium in Seattle sponsored by Boeing and the University of Washington [58].

In 1976, the US Army Material Command incorporated FTA into the Engineering Design Handbook, Design for Reliability [58]. The Reliability Analysis Center at Rome Laboratory, has published documents on FTA and reliability block diagrams since the 1960s [Chapter 6 (FTA) in MIL-HDBK-338B (Electronic Reliability Design Handbook)].

In 1998, the US Federal Aviation Administration (FAA) the FAA published Order 8040.4 establishing risk management policy and hazard analysis in a range of critical activities beyond aircraft certification, including air traffic control and modernization of the US National Airspace System. This led to the publication of the FAA System Safety Handbook, which describes the use of FTA in various types of formal hazard analysis [58].

Nowadays, the FTA methodology is so used in system safety and reliability engineering, and in all different fields of engineering.

The Fault tree (FT) method makes it possible to identify failures and the different scenarios that lead to the occurrence of a feared event (FA). It is a qualitative and quantitative graphic approach at the same time. Indeed, at first, it identifies the different combinations leading to the adverse event, and then, from these critical scenarios, we can quantify the probability of occurrence of this feared event.

This approach follows a deductive process; it starts from a single event to then identify the different causes possibly triggering this event. Graphically, the feared event appears at the top, it is the starting point for creating elementary events by tree structure. Fault tree

(FT) evaluates the state of a system using Boolean logic to combine lower-level events in this system. Table 1.1 shows some graphical symbols in FT. The operators in FT are inspired by the logic gates in digital electronics. The top and intermediate events represented by rectangles are progressively broken down into combinations of lower-level events until the reach of elementary basic events represented by circles. Basic events usually denote the failure of components while top event usually indicates the failure of a system.

Figure 1.18 gives FTs of all the RBDs in Figure 1.4.

Figure 1.14 is the FT of a series system. Where the failure of any component causes the failure of the whole system. Therefore, OR gate in FT corresponds to the series system in RBD. Figure 1.15 is the FT of a parallel system. Where only the failure of all three components causes the failure of the system. Therefore, AND gate in FT corresponds to the parallel system in RBD. Figure 1.16 is the FT of a k-out-of-n system. In such system, the failure of at least $(n - k + 1)$ component causes the failure of the system. Therefore, $(n - k + 1)$-out-of-n gate in FT corresponds to the k-out-of-n voter system in RBD. Figure 1.17 is the FT of the complex system in Figure 1.4. All the possible failure scenarios are listed as branches of the FT. The definition of minimum cut will be given in subsection 1.7.1.4.

If we want to calculate the reliability/ availability (probability that the system works) of the series system by using the FT in Figure 1.14, we must calculate the unreliability/unavailability (the probability that the system fails) of the system. In other words, we calculate the probability that the top event occurs. The top event represents here the failure of the whole system. The probability of the failure of the system is equal to the probability of the failure of C_1, C_2 or C_3, and it is presented as follows:

$$\begin{cases} P(\text{the whole system fails}) = P(C_1 \text{ fails or } C_2 \text{ fails or } C_3 \text{ fails}) = P(\overline{C_1} \cup \overline{C_2} \cup \overline{C_3}) \\ P(\text{the whole system works}) = 1 - P(\text{the whole system fails}) \end{cases}$$

1.7.1.3. Bayesian network

Bayesian network (BN) is a probabilistic graphical model used to represent a set of random variables and their conditional dependencies via a directed acyclic graph. It was developed by Judea Pearl [88] as a framework for representing and evaluating models under uncertainty [109, 46, 61].

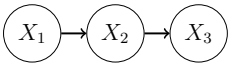


Figure 1.19: Series

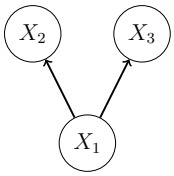


Figure 1.20: Diverging

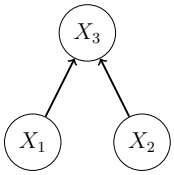


Figure 1.21: Converging

Figure 1.22: BNs of different structures.

During the last decades, the use of BN in risk and reliability analysis studies has increased. Weber *et al.* [120] presented a review on the application of BN to dependability, risk analysis and maintenance. In this paper, the authors claim that during the last few years, there is an increasing use of BN in dependability and risk analysis. They have also compare BN with three classical dependability and risk analysis methods: FTs, Markov chains and Petri nets, to demonstrate the benefits of BN. Langseth and Portinale [61] also list some arguments for the use of BNs in the reliability analysis (software reliability, fault finding systems, general reliability modeling, etc.). They present in the context challenges and open problems when using BN such as building BNs from expert input or using continuous variables in BNs. Akhtar and Utne [3] studied the human fatigue's effect on the risk of maritime ship accident using BN. Khakzad *et al.* [52] developed a BN method to conduct quantitative risk analysis of drilling operations. BNs are used to develop accident scenarios. Zhang *et al.* [128] estimated the navigational risk of the Yangtze River using the formal safety assessment and BN. In this context, BNs were used to model and evaluate accidents due to different factors. Sousa and Einstein [106] presented a methodology to assess systematically and manage the risks associated with tunnel construction using BN. Goulding *et al.* [39] proposed a BN model to assess the public health risk associated with wet weather sewer

overflows. The uncertainty inherent in sewer overflow events and subsequent impacts were taken into account through the use of probabilities.

A brief recall about Bayesian networks is provided hereafter. [109, 61, 57].

Known as an inference probabilistic method, BN is composed of nodes, arcs and conditional probability tables (CPTs). Nodes represent random variables. They may be observable elements, unknown parameters, or propositions. Arcs represent direct dependencies between linked variables. The strength of these dependencies is quantified by conditional probabilities. Nodes that are not directly connected represent variables that are conditionally independent. CPTs are used to specify the conditional probabilities among random variables. BN is used to estimate the posterior probability of unknown variables given other variables, through a process known as probabilistic reasoning.

Let $U = \{X_1, X_2, \ldots, X_n\}$ be a set of n random variables. If there exists a directed path from X_i to X_j, we call X_i a parent of X_j, and X_j a descendant of X_i. Figure 1.22 shows three BNs of different structures. In Figure 1.19, X_1 is the parent of X_2, X_2 is the parent of X_3, written $pa(X_2) = \{X_1\}$, $pa(X_3) = \{X_2\}$. In Figure 1.20, X_1 is the parent of X_2 and X_3, written $pa(X_2) = \{X_1\}$, $pa(X_3) = \{X_1\}$. In Figure 1.21, X_1 and X_2 are parents of X_3, written $pa(X_3) = \{X_1, X_2\}$.

The topology of a BN represents which variables are conditionally independent given another variable. For example, in Figure 1.20, X_2 is conditionally independent of X_3, given X_1 if $P(X_2|X_3, X_1) = P(X_2|X_1)$.

The advantage of BNs is that they provide a compact representation of the joint probability distribution of the variables. This probability can be expressed as a product of the conditional distributions of each node given its parents in the graph. Let $pa(X)$ denote the parents of node X, the joint distribution $P(U)$ has the following form:

$$P(U) = \prod_{X \in U} P(X|pa(X)) \tag{1.30}$$

For example, the joint probability of the model in Figure 1.19 is

$$P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$$

the joint probability of the model in Figure 1.20 is

$$\begin{aligned} P(X_1, X_2) &= P(X_2|X_1)P(X_1) \\ P(X_1, X_3) &= P(X_3|X_1)P(X_1) \end{aligned} \tag{1.31}$$

the joint probability of the model in Figure 1.21 is

$$P(X_1, X_2, X_3) = P(X_3|X_1, X_2)P(X_1)P(X_2)$$

This reasoning is called probabilistic reasoning. It consists in instantiating the input variables and propagating their effect through the network to update the probability of the variables of interest. The propagating procedure is based on Bayes' theorem and the structure of dependencies of the network.

As BNs often represent causal relationship like $X \rightarrow Y$, where X is a cause of Y and Y is an observable effect of X. The posterior probability distribution $P(X|Y = y)$ given the observation $Y = y$ can be computed using the prior distribution $P(X)$ and the conditional probability distribution $P(Y|X)$. Bayes' rule is expressed in the following form

$$P(X|Y = y) = \frac{P(Y = y|X)P(X)}{P(Y = y)} \tag{1.32}$$

where $P(Y = y) = \sum_x P(Y = y|X = x)P(X = x)$.

As described above, a BN contains two parts: a qualitative part consisting of a directed acyclic graph, and a quantitative part consisting of a joint probability distribution that factorizes into a set of conditional probability distributions governed by the structure of the directed acyclic graph.

MSBNx (Microsoft Bayesian Network Editor) is a component-based toolbox for creating, assessing, and evaluating BNs. As a simple example, we use MSBNx to construct the BN of the system in Figure 1.21.

Suppose all three random variables are binary and the probabilities of input variables are

$$\begin{aligned} P(X_1 = Yes) &= 0.7 \qquad P(X_1 = No) = 0.3 \\ P(X_2 = Yes) &= 0.8 \qquad P(X_2 = No) = 0.2 \end{aligned}$$

CPTs can be determined by experiments. To simplify the example, we set the CPT of all

Parent Node(s)		X3		
X1	X2	Yes	No	bar charts
Yes	Yes	1,0	0,0	
	No	0,6	0,4	
No	Yes	0,8	0,2	
	No	0,0	1,0	

Figure 1.23: CPT

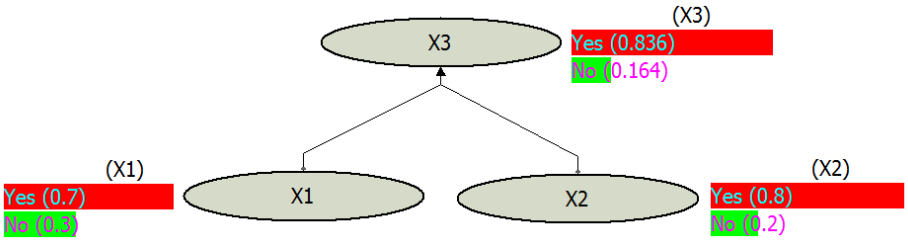


Figure 1.24: BN of the system in Figure 1.21[91].

three variables as shown in Figure 1.23. It means

$$P(X_3 = Yes|X_1 = Yes, X_2 = Yes) = 1 \qquad P(X_3 = No|X_1 = Yes, X_2 = Yes) = 0$$
$$P(X_3 = Yes|X_1 = Yes, X_2 = No) = 0.6 \qquad P(X_3 = No|X_1 = Yes, X_2 = No) = 0.4$$
$$P(X_3 = Yes|X_1 = No, X_2 = Yes) = 0.8 \qquad P(X_3 = No|X_1 = No, X_2 = Yes) = 0.2$$
$$P(X_3 = Yes|X_1 = No, X_2 = No) = 0 \qquad P(X_3 = No|X_1 = No, X_2 = No) = 1$$

Usually, the outputs of a BN are the posterior probabilities of variables of interest. In this example, X_3 is the variable of interest. The probability of X_3 assessed by MSBNx is given in Figure 1.24. To prove the correctness of this result given by the software, we calculate the probability of X_3 using the law of total probability. We have

$$P(X_3 = Yes) = 0.7 * 0.8 * 1 + 0.7 * 0.2 * 0.6 + 0.3 * 0.8 * 0.8 + 0.3 * 0.2 * 0 = 0.836$$
$$P(X_3 = No) = 0.7 * 0.8 * 0 + 0.7 * 0.2 * 0.4 + 0.3 * 0.8 * 0.2 + 0.3 * 0.2 * 1 = 0.164$$

1.7.1.4. Kaufmann reliability network

Kaufmann reliability networks are an interesting method to calculate the reliability of a system [51, 78, 54]. They are used in the communication studies or energy distribution networks [102]. This graphical representation is a modeling tool for many problems where the use of reliability block diagrams (RBD) is not appropriate. Kaufmann reliability networks can be considered as a form of extension of RBDs. Thus, these reliability networks provide a simple way of computing the reliability of the system. Reliability networks make it possible to represent the structure and the connections in a graphical system by expressing the relationships between the components.

Kaufmann reliability network use concepts of graph theory as a tool to analyze systems dependability. In the following, we first recall the basics of graph theory and then show how it applies to reliability networks. Note that, in our case,these networks can represent systems in a broad sense (logistics transport, telecommunications, IT, embedded systems, etc.). A network is a system in which entities communicate with each other by sending a flow, from a source node to a target node. It is similar to an oriented graph, whose nodes and arcs can fail according to a certain probability.

◇ **Definitions**

A reliability network R is a oriented graph $G = (V, E)$ whose arcs represent the components $E = \{C_1, C_2, \ldots, C_n\}$ and the vertexes $V = \{V_1, V_2, \ldots, V_n\}$ represent the nodes. This graph G is without loop, in which two vertexes $S \in V$ and $T \in V$ are distinguished and called respectively "origin" and "end". Each arc $U_i \in U$ is noted $(V_j V_k)_{C_l}$ such as V_j and $V_k \in V$ and $C_l \in E$ with U all the arcs in the graph [95].

$\Omega : U \longrightarrow V \times V$ corresponds to each arc in the graph, the couple of its extremities. The same end pair can represent several arcs such as each arc corresponds to a component.

$\Delta : V \times V \longrightarrow E$ corresponds to each arc in the graph represented by its initial and final ends a components. Several arcs can represent the same component.

Figure 1.25 gives an example of a connection diagram (a) and its associated reliability network (b). The locations of nodes A and B on the connection diagram are represented by blue dots and the origin nodes "S" and end nodes "T" by black dots.

The set of nodes is given by $V = \{S, A, B, T\}$ and the set of components is $E = \{C_1, C_2, C_3, C_4, C_5\}$.

$$\begin{cases} \Omega : E \longrightarrow V \times V \\ \Omega(C_1) = (S, A) \end{cases} \tag{1.33}$$

$$\begin{cases} \Delta : V \times V \longrightarrow E \\ \Delta(A, T) = C_4 \end{cases} \tag{1.34}$$

We can notice that the same torque (S, A) for example, represents two arcs $(SA)_{C_1}$ and $(SA)_{C_2}$. Thus, there are two arcs represented by the pairs (S, A) and (S, B) which are generated by the same component C_2.

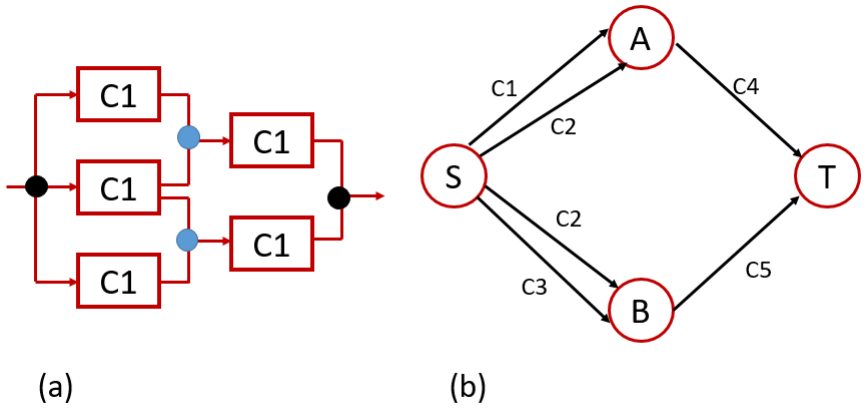


Figure 1.25: Example of a connection diagram (a) and its reliability network (b)

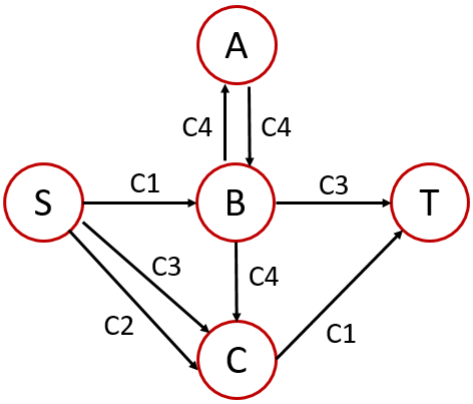


Figure 1.26: Example of a reliability network

A path C of a reliability network is a subset of an arc $U' \subset U$ connecting the origin point S to the end point T. A path C is said to be minimum when no subset $U'' \subset U$ cannot present a path, such that U' is the set of arcs of the path C.

From Figure 1.26, we deduce the following paths:

$$\begin{cases} C_1 = \{(SB)_{C_1}, (BA)_{C_4}, (AB)_{C_4}, (BT)_{C_3}\} \\ C_2 = \{(SB)_{C_1}, (BT)_{C_3}\} \\ C_3 = \{(SB)_{C_1}, (BC)_{C_4}, (CT)_{C_1}\} \\ C_4 = \{(SC)_{C_3}, (CT)_{C_1}\} \\ C_5 = \{(SC)_{C_2}, (CT)_{C_1}\} \end{cases} \tag{1.35}$$

The paths C_2, C_3, C_4 and C_5 are minimum paths of the reliability network in Figure 1.26, whereas the path C_1 is not a minimum path because all the arcs of the path C_2 are included

in all the arcs of the path C_1.

A cut D of an reliability network is a subset of components $E' \subset E$. Removing these components would disconnect both extremities of the network. And therefore, there would be no path from origin point S to end point T.
A cut D is said to be minimum when no subset of components $E'' \subset E'$ presents a cut, such that E' is the set of components of the cut D.

In the example in Figure 1.26, $\{\overline{C_1}, \overline{C_2}, \overline{C_3}\}$ and $\{\overline{C_1}, \overline{C_3}\}$ are network cuts but they are not minimum, whereas $\{\overline{C_1}\}$ and $\{\overline{C_2}, \overline{C_3}, \overline{C_4}\}$ are minimum cuts.

◇ **RAMS assessment at the system level**

The reliability of a system can be deduced from the reliability of components and the structure of the system. When the reliability of components is given, we can use two methods to assess the reliability of the system.

• ***Minimum path-sets***

A path-set is defined as a set of components whose operation guarantees the correct operation of a system. A minimum path-set is a path-set which does not contain other path-sets.

Let $\mathcal{P}_j$ denote the event that the components in the minimum path-set p_j are working. $\mathcal{A}_p$ is the collection of minimum path-sets. The reliability of the system is:

$$R_s = P(\cup_{\mathcal{P}_j \in \mathcal{A}_p} \mathcal{P}_j) \tag{1.36}$$

Using the inclusion-exclusion principle, the reliability of a complex system can be developed as follows:

$$R_s = \sum_{\mathcal{I} \subseteq \mathcal{A}_p} (-1)^{|\mathcal{I}|+1} P(\cap_{\mathcal{P}_j \in \mathcal{I}} \mathcal{P}_j) \tag{1.37}$$

with $|\mathcal{I}|$ is the cardinal the cardinal of $\mathcal{I}$ the subset of events. Eq. 1.37 is obtained from the theorem of Sylvester-Poincaré. For example, the collection of minimum path-sets of the complex system in Figure 1.7.1 is $\{C1C4, C2C4, C2C5, C3C5\}$. Based on Eq. 1.37, the reliability of the complex system is obtained as $R_s = R1R4 + R2R4 + R2R5 + R3R5 - R1R2R4 - R2R4R5 - R2R3R5 - R1R3R4R5 + R1R2R3R4R5$.

• ***Minimum cut-sets***

A cut-set is defined as a set of components whose failure guarantees the failure of a system. A minimum cut-set is a cut-set which does not contain other cut-sets.

Let $\mathcal{C}_j$ denote the event that the components in the minimum cut-set c_j are failed. $\mathcal{A}_c$ is the collection of minimum cut-sets. The unreliability of the system is

$$\overline{R}_s = P(\cup_{\mathcal{C}_j \in \mathcal{A}_c} \mathcal{C}_j) \tag{1.38}$$

Using the inclusion-exclusion principle (see chapter 3), the reliability of a complex system is developed as

$$R_s = 1 - \sum_{\mathcal{I} \subseteq \mathcal{A}_c} (-1)^{|\mathcal{I}|+1} P(\cap_{\mathcal{C}_j \in \mathcal{I}} \mathcal{C}_j) \tag{1.39}$$

For example, the collection of minimum cut-sets of the complex system in Figure 1.7.1 is $\{\overline{C1}\,\overline{C2}\,\overline{C3},\ \overline{C4}\,\overline{C5},\ \overline{C1}\,\overline{C2}\,\overline{C5},\ \overline{C2}\,\overline{C3}\,\overline{C4}\}$. Based on Eq. 1.39, the reliability of the complex system is obtained as $R_s = R1R4 + R2R4 + R2R5 + R3R5 - R1R2R4 - R2R4R5 - R2R3R5 - R1R3R4R5 + R1R2R3R4R5$.

1.8 Conclusion

Dependability is the main subject of this thesis. Particularly, the evaluation of the system's availability is of main concern. In this chapter we exposed the main definitions, concepts and methods related to dependability. We also presented the different types of systems (series, parallel, complex systems, etc.). Even if the presented methods that concern the structures of the system are simple to apply, however when having large systems (in terms of components and states) their use become limited. In the following chapters, we focus on complex systems and consider the case where uncertainty is present. Hence, this chapter also included a brief explanation of uncertainties and the difference between aleatory and epistemic uncertainty. The main novelty of this thesis is to model uncertainties in terms of intervals. Thus, the following chapter, will be devoted to present basic definitions of interval analysis and its operators, also an explanation of a technique used in this domain "the technique of contractors" will be given, so that we can use it in our proposed methodology in chapter 4.

Chapter 2

Interval analysis

2.1 Introduction

Mathematicians developed a method called interval computation, interval analysis, interval mathematics, or interval arithmetic [79, 80, 81]. It is an approach to putting bounds on rounding errors and measurement errors in mathematical computation and thus developing numerical methods that yield reliable results. Each value is represented by a range of possibilities. Our methods in Part *II* are based on interval analysis.
Before using interval analysis in these methods, we need to introduce its main concepts. First, we start with a brief history about interval analysis. Then, we explain the basic concept of interval analysis. Later, we cite all the definitions and the operations in interval analysis. Contractors, operators introduced in interval analysis, will be used in our methods. For this reason, we explain their concept in this chapter.

2.2 History of interval analysis

In this section, we give a historical view of how interval analysis was developed. For sure, we cannot give all the details concerning this development. However, we try to list some of the marks and works in the field of interval analysis.

Interval analysis is not recently introduced in mathematics; it has appeared many times under different names before. In the third century BC, Archimedes calculated the lower and the upper bounds: $223/71 < \pi < 22/7$.

Calculation with intervals and its rules were published in a 1931 work by Rosalind Cecily Young [125]. In 1951, a textbook on linear algebra by Paul Dwyer [32] was published

where he works to improve the reliability of digital systems; he used intervals to measure rounding errors associated with floating-point numbers.

Later in 1958, Teruo Sunaga published a paper on interval algebra in numerical analysis [108]. In this paper, not only the algebraic rules for the operations with intervals are mentioned but also a systematic investigation of the rules which they achieve.

Modern interval arithmetic was really appears with the book "Interval Analysis" by Ramon E. Moore in 1966 [79, 81]. He had the idea in 1958, in the next year he published an article about computer interval arithmetic [45]. It starts with a simple principle that provided a general method for automated error analysis, not just errors resulting from rounding.
Moore's book was inspired from his Ph.D. thesis [82]. Therefore, he was mainly concentrated on bounding solutions of initial value problems for ordinary differential equations, however it contained also a lot of general ideas.

Also in 1956, to calculate with intervals Mieczyslaw Warmus suggested formulas, but Moore found the first non-trivial applications.

In the next twenty years, groups of German researchers continued in this field Götz Alefeld and Ulrich Kulisch at the University of Karlsruhe and at the Bergische University of Wuppertal. In the 1960s, Eldon R. Hansen extend intervals for linear equations and then provided important contributions to global optimization (including Hansen's method), perhaps the most used interval algorithm [44].

In 1988, Fortran-based software for reliable solutions for initial value problems using ordinary differential equations was developed by Rudolf Lohner [72].

Since 1990, the journal Reliable Computing (Interval Computations) has been published. The editor R. Baker Kearfott, has contributed to the unification of notation and terminology used in interval arithmetic, in addition to his work on global optimization.

Recently, the estimation of preimages of parametrized functions and to robust control theory is the main focused work (by the COPRIN working group of INRIA in Sophia Antipolis in France) [56].

2.3 What is interval analysis?

The mathematical operations on floating points instead of real numbers make an accumulation of errors due to rounding. We will find at the end that the obtained result can be far from the expected result. Thanks to Ramon Moore's work in [79], the solution is described as follows. On the one hand, the reals are represented by intervals bounded by floating with a fixed number of decimals, for example $\pi = 3.14159\ldots$, will be represented by $[3.1415, 3.1416]$ because $\pi \in [3.1415, 3.1416]$.
On the other hand, operation tools with intervals represent the result of a mathematical function by an interval that contains it in a guaranteed way.

The aim of interval analysis is to create methods that cope with all kind of imprecision that blocks classical numerical techniques from providing reliable results [20]. In real life problems, we can model imprecision with rounding errors as well as data uncertainties. The main idea is to fix intervals where the range of all possible error made belongs to them, in any low-level computation. Therefore, computations are performed with the so-called interval arithmetic, that takes intervals instead of real values, e.g.,

$$[2, 3] + [1, 4] = [3, 7]$$

Of course, the representation by intervals does not have only advantages. One of the serious problems, is the overestimation of the uncertainty bounds of the obtained result by interval computing [20].

For systems where uncertainties occupied, interval analysis is a promising methodology.

2.4 Basic terms and definitions

An interval real $[x]$ is a closed subset of $\mathbb{R}$. Although various other types of intervals (open, half-open) appear throughout mathematics, our work will center primarily on closed intervals. We will keep the notation $[x]$ to any interval.

Endpoint Notation

An interval is defined as:

$$[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \overline{x}\}, \tag{2.1}$$

where $\underline{x}$ and $\overline{x}$ refer respectively to the lower bound and the upper bound of $[x]$.

The lower bound is defined as:

$$\underline{x} \triangleq sup\left\{a \in \mathbb{R} | \forall x \in [x], a \leq x\right\}, \tag{2.2}$$

Its upper bound, is defined as:

$$\overline{x} \triangleq inf\left\{b \in \mathbb{R} | \forall x \in [x], x \leq b\right\}, \tag{2.3}$$

$\underline{x}$ is the largest number on the left of $[x]$ and $\overline{x}$ is the smallest number on the right. Example, if $[x] = [-2, 5]$ then $\underline{x} = -2$ and $\overline{x} = 5$; if $[x] =]-\infty, +\infty[$ then $\underline{x} = -\infty$ and $\overline{x} = +\infty$.

Interval equality

Two intervals $[x]$ and $[y]$ are said to be *equal*, if they are the same sets. This happens when their corresponding endpoints are equal:

$$[x] = [y] \text{ if } \underline{x} = \underline{y} \text{ and } \overline{x} = \overline{y} \tag{2.4}$$

Width and midpoint of interval

The *width* of an interval is defined as:

$$w([x]) \triangleq \overline{x} - \underline{x} \tag{2.5}$$

The *width* of $[-2, 5]$ is $w([-2, 5]) = 7$.
The *midpoint* (or *center*) of any bounded interval $[x]$ is defined as:

$$mid([x]) \triangleq \frac{\underline{x} + \overline{x}}{2} \tag{2.6}$$

The *center* of $[-2, 5]$ is $mid([-2, 5]) = \frac{-2+5}{2} = 1.5$.

Degenerate Intervals

$[x]$ is degenerate if $\underline{x} = \overline{x}$. A single real number x belongs to such interval. We identify a degenerate interval $[x, x]$ with the real number x. In this sense, we can write such equations as:

$$0 = [0, 0] \tag{2.7}$$

Intersection, Union, and Interval Hull

The set-theoretic operations are also applied to intervals. The *intersection* of two intervals $[x]$ and $[y]$ is empty if either $\overline{y} < \underline{x}$ or $\overline{x} < \underline{y}$. Let $\emptyset$ presents the empty set, we will have:

$$[x] \cap [y] = \emptyset \tag{2.8}$$

Which means that $[x]$ and $[y]$ have no points in common.
The *intersection* of two intervals $[x]$ and $[y]$, defined by

$$[x] \cap [y] \triangleq \{z \in \mathbb{R} | z \in [x] \text{ and } z \in [y]\} \tag{2.9}$$

in other way

$$\begin{aligned} [x] \cap [y] &= \{z : z \in [x] \text{ and } x \in [y]\} \\ &= \left[max\left\{\underline{x}, \underline{y}\right\}, min\left\{\overline{x}, \overline{y}\right\}\right] \end{aligned} \tag{2.10}$$

which is also an interval. In general, this is not the case for their *union*, it is not always an interval

$$[x] \cup [y] \triangleq \{z \in \mathbb{R} | z \in [x] \text{ or } z \in [y]\} \tag{2.11}$$

in a another way

$$\begin{aligned} [x] \cup [y] &= \{z : z \in [x] \text{ or } x \in [y]\} \\ &= \left[min\left\{\underline{x}, \underline{y}\right\}, max\left\{\overline{x}, \overline{y}\right\}\right] \end{aligned} \tag{2.12}$$

However, the *interval hull* of two intervals, defined by

$$[x] \underline{\cup} [y] \triangleq [[x] \cup [y]] \tag{2.13}$$

is an interval and can be used in interval computations. We have $[x] \cup [y] \subset [x] \underline{\cup} [y]$.
Example, the intersection $[2, 5] \cap [3, 7] = [3, 5]$ however its union is $[2, 5] \cup [3, 7] = [2, 7]$.

Importance of Intersection

Intersection plays an important role in interval analysis. Suppose that there are two intervals that contain a result of interest, then their intersection also contains the result.

Example, if two people make independent measurements of the same measure *m*. One finds $m = 5.4$ with a measurement error less than 0.2. The other finds that $m = 5.2$ with an error less than 0.2. We can represent the two obtained values of *m* respectively by two intervals $[m_1] = [5.2, 5.6]$ and $[m_2] = [5, 5.4]$. Since *m* depends on the two measurements, it depends also on $[m_1] \cap [m_2] = [5.2, 5.4]$.
The empty intersection means that at least one of the measurements is wrong.

Order Relations for Intervals

Real numbers are ordered by the relation $<$, e.g. $3 < 5, -5 < 2$. This relation is transitive, which means that if $a < b$ and $b < c$, so $a < c, \forall a, b, c \in \mathbb{R}$. For intervals we can define the same thing and we use the same symbol, with the following definition:

$$[x] < [y] \text{ means that } \overline{x} < \underline{y} \tag{2.14}$$

Set inclusion is another transitive order relation for intervals:

$$[x] \subseteq [y] \text{ if } \underline{y} \leq \underline{x} \text{ and } \overline{x} \leq \overline{y} \tag{2.15}$$

That means that $[y]$ contains $[x]$.

$[x]$ and $[y]$ are overlapping intervals when $[x]$ is not contained in $[y]$, nor $[y]$ is contained in $[x]$. However, $[x] \cap [y]$ is contained in $[x]$ and $[y]$.

Example: $[3, 8]$ and $[4, 10]$ the two intervals do not contain to each other, however their intersection $[4, 8]$ is contained in both of them.

2.5 Interval computation

In this section, we will define the basic arithmetic operations between intervals, these definitions are taken from Moore's book [81]. Computing with intervals is computing with sets. Which means, if we add two intervals, the resulting interval is a set containing the sums of all pairs of numbers, one from each of the two initial sets [81].
Let us define the four arithmetic operations that are used the most, then we detail each one apart:

- The *sum* of two intervals $[x]$ and $[y]$ is the set:

$$[x] + [y] = \{x + y : x \in [x], y \in [y]\} \tag{2.16}$$

- The *difference* of two intervals $[x]$ and $[y]$ is the set:

$$[x] - [y] = \{x - y : x \in [x], y \in [y]\} \tag{2.17}$$

- The *product* of $[x]$ and $[y]$ is given by:

$$[x].[y] = \{xy : x \in [x], y \in [y]\} \tag{2.18}$$

- The *quotient* $[x]/[y]$ is defined as:

$$[x]/[y] = \{x/y : x \in [x], y \in [y], 0 \notin [y]\} \tag{2.19}$$

In general, we can summarize the above definitions by writing:

$$[x] \diamond [y] = \{x \diamond y : x \in [x], y \in [y]\} \tag{2.20}$$

Where $\diamond$ refers for any of the four binary operations $\{+, -, \times, \div\}$.

2.5.1 Interval addition

Since

$$x \in [x] \text{ means that } \underline{x} \leq x \leq \overline{x} \tag{2.21}$$

and

$$y \in [y] \text{ means that } \underline{y} \leq y \leq \overline{y} \tag{2.22}$$

the numerical sums $x + y \in [x] + [y]$ must satisfy

$$\underline{x} + \underline{y} \leq x + y \leq \overline{x} + \overline{y} \tag{2.23}$$

We will have:

$$[x] + [y] = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right] \tag{2.24}$$

Example: let $[x] = [1, 2]$ and $[y] = [-1, 3]$, then $[x] + [y] = [1 + (-1), 2 + 3] = [0, 5]$.
It is not the same as $[x] \cup [y] = [-1, 3]$. The sum of two intervals it doesn't mean their union.

2.5.2 Interval subtraction

The sum $[x] + [y]$ is expressed in eq. 2.24 in terms of the endpoints of $[x]$ and $[y]$. Derived for the remaining arithmetic operations we can write similar expressions. For subtraction we add the inequalities:

$$\underline{x} \leq x \leq \overline{x} \text{ and } -\overline{y} \leq y \leq -\underline{y} \tag{2.25}$$

to get

$$\underline{x} - \overline{y} \leq x - y \leq \overline{x} - \underline{y} \tag{2.26}$$

It follows that

$$[x] - [y] = \left[\underline{x} - \overline{y}, \overline{x} - \underline{y}\right] \tag{2.27}$$

Note that

$$[x] - [y] = [x] + (-[y]) \tag{2.28}$$

Case	$\underline{xy}$	$\overline{xy}$
$0 \leq \underline{x}$ and $0 \leq \underline{y}$	$\underline{x}.\underline{y}$	$\overline{x}.\overline{y}$
$\underline{x} < 0 < \overline{x}$ and $0 \leq \underline{y}$	$\underline{x}.\overline{y}$	$\overline{x}.\overline{y}$
$\overline{x} \leq 0$ and $0 \leq \underline{y}$	$\underline{x}.\overline{y}$	$\overline{x}.\underline{y}$
$0 \leq \underline{x}$ and $\underline{y} < 0 < \overline{y}$	$\underline{x}.\overline{y}$	$\overline{x}.\overline{y}$
$\overline{x} \leq 0$ and $\underline{y} < 0 < \overline{y}$	$\underline{x}.\overline{y}$	$\underline{x}.\underline{y}$
$0 \leq \underline{x}$ and $\overline{y} \leq 0$	$\overline{x}.\underline{y}$	$\underline{x}.\overline{y}$
$\underline{x} < 0 < \overline{x}$ and $\overline{y} \leq 0$	$\overline{x}.\underline{y}$	$\underline{x}.\underline{y}$
$\overline{x} \leq 0$ and $\overline{y} \leq 0$	$\overline{x}.\overline{y}$	$\underline{x}.\underline{y}$
$\underline{x} < 0 < \overline{x}$ and $\underline{y} < 0 < \overline{y}$	$min\left\{\underline{x}\overline{y}, \overline{x}\underline{y}\right\}$	$max\left\{\underline{x}\underline{y}, \overline{x}\overline{y}\right\}$

Table 2.1: Endpoint formulas for interval product

where

$$-[y] = [-\overline{y}, -\underline{y}] = \{y : -y \in [y]\} \tag{2.29}$$

Example: let $[x] = [-1; 1]$ and $[y] = [0, 2]$, then

$$-[y] = [-2, 0] \text{ and } [x] - [y] = [x] + (-[y]) = [-3, 1]$$

2.5.3 Interval multiplication

The product $[x].[y]$ of two intervals $[x]$ and $[y]$ is given by:

$$[x].[y] = [\min S, \max S], \text{ where } S = \left\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right\} \tag{2.30}$$

Example, if $[x] = [-1, 1]$ and $[y] = [0, 2]$, then:

$$S = \{-1.0, -1.2, 1.0, 1.2\} = \{0, -2, 0, 2\}$$

and

$$[x].[y] = [\min S, \max S] = [-2, 2]$$

$$2[y] = [2, 2].[0, 2] = [0, 4]$$

The product of intervals is given in terms of the minimum and maximum of four products of endpoints. However, if we test the sign of the endpoints $\underline{x}, \overline{x}, \underline{y}, \overline{y}$, the formula for the endpoints of the interval product can be broken into nine special cases. Table 2.1 presents all the possible cases.

$[x] + [y]$	$= [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$
$[x] - [y]$	$= [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$
$[x] \times [y]$	$= [min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})]$
$[x] \div [y]$	$= [x] \times [1/\overline{y}, 1/\underline{y}], 0 \notin [y]$

Table 2.2: Basics interval operations

2.5.4 Interval division

Like when dealing with real numbers, division can be done via multiplication by the reciprocal of the second operand. From eq. 2.19, we can write down:

$$[x]/[y] = [x].(1/[y]) \tag{2.31}$$

where

$$1/[y] = \{y : 1/y, y \in [y]\} = [1/\overline{y}, 1/\underline{y}] \tag{2.32}$$

with $0 \notin [y]$.

Example: To solve the equation $ax = b$ we can use the division, with *a* and *b* belongs to two intervals $[a]$ and $[b]$, respectively. We find that *x* must belong to $[b]/[a]$.

2.6 Properties of intervals

We introduced the definitions of the basic interval arithmetic operations [81], Table 2.2 summarizes the operations that we will use later. These definitions lead to a certain familiar looking algebraic properties. In this section, we present some of these properties.

2.6.1 Commutativity and associativity

Interval sum and interval product multiplication are commutative and associative; we have for any three intervals $[x], [y]$ and $[z]$:

$$\begin{aligned} [x] + [y] = [y] + [x],\ [x] + ([y] + [z]) = ([x] + [y]) + [z]; \\ [x].[y] = [y].[x],\ [x].([y].[z]) = ([x].[y]).[z] \end{aligned} \tag{2.33}$$

2.6.2 Additive and multiplicative identity elements

In the system of intervals, the degenerate intervals $[0]$ and $[1]$ are additive and multiplicative identity elements. For any $[x]$, we have:

$$\begin{cases} [0] + [x] = [x] + [0] = [x]; \\ [0].[x] = [x].[0] = [0]; \\ [1].[x] = [x].[1] = [x]. \end{cases} \tag{2.34}$$

2.6.3 Nonexistence of inverse elements

In the system of intervals, $-[x]$ is not an additive inverse for $[x]$. We have:

$$[x] + (-[x]) = [\underline{x}, \overline{x}] + [-\overline{x}, -\underline{x}] = [\underline{x} - \overline{x}, \overline{x} - \underline{x}], \tag{2.35}$$

It is equal to $[0, 0]$ only if $\underline{x} = \overline{x}$. If the width of $[x]$ is not equal to zero, then:

$$[x] - [x] = \begin{cases} [\underline{x}, \overline{x}] - [\underline{x}, \overline{x}] \\ = [\underline{x} - \overline{x}, \overline{x} - \underline{x}] \\ = [-(\overline{x} - \underline{x}), \overline{x} - \underline{x}] \\ = (\overline{x} - \underline{x})[-1, 1] \\ = w([x])[-1, 1] \end{cases} \tag{2.36}$$

Thus,

$$[x] - [x] = w([x])[-1, 1] \tag{2.37}$$

The same thing for $[x]/[x] = 1$ only if $w([x]) = 0$. Usually, we have:

$$[x]/[x] = \begin{cases} [\underline{x}/\overline{x}, \overline{x}/\underline{x}] \text{ if } 0 < \underline{x}, \\ [\overline{x}/\underline{x}, \underline{x}/\overline{x}] \text{ if } \overline{x} < 0. \end{cases} \tag{2.38}$$

Degenerate intervals only have additive or/and multiplicative inverses.

2.6.4 Sub-distributivity

The distributive law of ordinary arithmetic,

$$x(y + z) = xy + xz \tag{2.39}$$

is not applied for intervals.

Lets take $[x] = [1, 2]$, $[y] = [1, 1]$ and $[z] = -[1, 1]$:

$$\begin{cases} [x]([y]+[z]) = [1,2].([1,1]-[1,1]) \\ = [1,2].[0,0] \\ = [0,0] \end{cases}$$

and we have

$$\begin{cases} [x][y]+[x][z] = [1,2][1,1]-[1,2][1,1] \\ = [1,2]-[1,2] \\ = [-1,1] \end{cases}$$

We can define the sub-distributive law as follows:

$$[x]([y]+[z]) \subseteq [x][y]+[x][z] \tag{2.40}$$

2.6.5 Cancellation

The cancellation law holds for interval addition.

$$[x]+[z] = [y]+[z] \Rightarrow [x] = [y] \tag{2.41}$$

But, the multiplicative cancellation does not hold in interval arithmetic; which means:

$$[z][x] = [z][y] \nRightarrow [x] = [y] \tag{2.42}$$

2.7 Interval matrices

An *interval matrix* is a matrix whose elements are interval numbers. Here an example of interval matrix:

$$[A] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} [0,2] & [-2,2] \\ [1,4] & [3,6] \end{bmatrix} \tag{2.43}$$

If *B* is a matrix with real elements B_{ij} such that $B_{ij} \in A_{ij}$ for all *i* and *j*, with A_{ij} are the elements of the interval matrix [A], then we write $B \in [A]$.

2.7.1 Matrix norm, width, and midpoint

The matrix norm for an interval matrix [A] is an extension of the maximum row sum norm for real matrices. It is given by:

$$\|[A]\| = max_i \sum_j |A_{ij}| \text{ with } |A_{ij}| = max\left(|\underline{A_{ij}}|, |\overline{A_{ij}}|\right) \tag{2.44}$$

If a real matrix *B* belongs to the interval matrix [A], then $\|B\| \leq \|[A]\|$.

The width $w([A])$ of a matrix [A], is defined by:

$$w([A]) = max_{i,j} w(A_{ij}) \tag{2.45}$$

The midpoint of the interval matrix [A] is the real matrix $m([A])$ where its elements are the midpoints of the corresponding interval elements of [A], $m([A]) \in [A]$

$$(m([A]))_{ij} = m(A_{ij}) \tag{2.46}$$

For the matrix in 2.43, we have:

$$\begin{aligned} \|[A]\| &= max\left\{|[0,2]| + |[-2,2]|, |[1,4]| + |[3,6]|\right\} \\ &= max\left\{2+2, 4+6\right\} \ = 10 \end{aligned}$$

$$\begin{aligned} w([A]) &= max\left\{w([0,2]), w([-2,2]), w([1,4]), w([3,6])\right\} \\ &= max\left\{2,3,4\right\} \\ &= 4 \end{aligned}$$

and

$$m([A]) = \begin{bmatrix} m([0,2]) & m([-2,2]) \\ m([1,4]) & m([3,6]) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2.5 & 4.5 \end{bmatrix}$$

2.7.2 Matrices computation

An interval matrix can be punctual if all its elements are punctual.
The lower bound of an interval matrix [A] is denoted by $\underline{A}$. It is the punctual matrix made up with the lower bounds of the interval components of [A]:

$$\underline{A} = \begin{bmatrix} \underline{a}_{11} & \cdots & \underline{a}_{1n} \\ \vdots & \ddots & \vdots \\ \underline{a}_{m1} & \cdots & \underline{a}_{mn} \end{bmatrix} \tag{2.47}$$

The same thing for the upper bound $\overline{A}$ of the interval matrix [A], is the punctual matrix made up with its interval upper bounds components:

$$\overline{A} = \begin{bmatrix} \overline{a}_{11} & \cdots & \overline{a}_{1n} \\ \vdots & \ddots & \vdots \\ \overline{a}_{m1} & \cdots & \overline{a}_{mn} \end{bmatrix} \tag{2.48}$$

If [A] and [B] are interval matrices, with $\diamond$ a binary operator, then:

$$\begin{cases} [A] \diamond [B] = [\{A \diamond B | A \in [A] \text{ and } B \in [B]\}] \\ [A] + [B] = ([a_{ij}] + [b_{ij}])_{1 \leq i \leq n, 1 \leq j \leq n} \\ [A] * [B] = (\sum_{k=1}^{n} [a_{ik}] * [b_{kj}])_{1 \leq i \leq n, 1 \leq j \leq n} \end{cases} \tag{2.49}$$

2.7.3 Interval matrices and dependency

When taking the product $[C] = [A][B]$ of an *m* by *p* interval matrix [A] and a *p* by *n* interval matrix [B], the *ij*th element of this product C_{ij} gives bounds on the range, for each *i* $1 \leq i \leq m$ and for each *j* $1 \leq j \leq n$,

$$C_{ij} = \left\{ M_{ij} = \sum_{k=1}^{p} P_{ik} Q_{kj} : P_{ik} \in A_{ik} \text{ and } Q_{kj} \in B_{kj} \text{ for } 1 \leq k \leq p \right\} \tag{2.50}$$

the obtained interval matrix [C] may contain point matrices *D*, that are not what we obtain by doing the multiplication of point matrices $P \in [A]$ and $Q \in [B]$.

The product of two interval matrices [A] and [B] gives:

$$[C] = [A][B] = \begin{bmatrix} [1,2] & [3,4] \end{bmatrix} \begin{bmatrix} [5,6] & [7,8] \\ [9,10] & [11,12] \end{bmatrix} = \begin{bmatrix} [32,52] & [40,64] \end{bmatrix} \tag{2.51}$$

If we take the matrix $D = \begin{bmatrix} 32 & 64 \end{bmatrix}$, it belongs to [C], but it does not correspond to the product of any point matrix $P \in [A]$ with any matrix $Q \in [B]$.
Actually, if we do the product of the lower bound matrix $\underline{A} = [1\ 3] \in [A]$ with the lower bounds of the first column of [B] $(5\ 9)^T$, we obtain the first element 32 of *D*. However, 64 the second element of *D* is obtained by doing the product of the upper bound matrix $\overline{A} = [2\ 4] \in [A]$ of [A] with the upper bounds $[8\ 12]^T$ of the second column of [B]. Interval arithmetic does not assume that the same point elements are chosen from the interval elements of [A] in forming the sets comprising the different columns of the product interval matrix [C]. This is why this type of interval dependency occurs [81]. The so-called dependency problem is a major obstacle to the application of interval arithmetic (example: $[x]^2 \neq [x].[x]$).

2.8 Constraint satisfaction problems and contractors

Constraint satisfaction is the process of finding a solution to a set of constraints that impose conditions that the variables must satisfy [112]. Therefore, the solution will be a set of values for the variables with satisfaction of all constraints, that is a point in the feasible region.

The technique used in constraint satisfaction to find the solution, depends on the kind of constraints. This technique use operators called *contractors*, that is why we call it *the technique of contractors*.

The technique of contractors has been developed to be one of the important part in *Interval Analysis* domain [49]. This technique helps to contract an interval $[x]$ into a smaller one $[x']$.

2.8.1 History of contractors

Constraint satisfaction antedates 1965 [36]. The real world problems, like workforce scheduling, that we now identify as constraint satisfaction, have always been with us.

The 8-queens game's problem, which preoccupied so many of the early constraint satisfaction researchers in artificial intelligence, it have been proposed by the chess player Max Bazzel in 1848. Mythology claims that a form of backtrack search, a powerful search paradigm that has become a central tool for constraint satisfaction, was used by Theseus in the labyrinth in Crete.

In recreational mathematics in the nineteenth century, they used the backtrack search. It was a subject of study as computer science and operations research emerged as academic disciplines after World War *II*. Bitner and Reingold [7] credit Lehmer with first using the term "backtrack" in the 1950's [43]. Many different forms of constraint satisfaction and propagation appeared in the 1960's, in the computer science literature [17, 18, 34, 74].

The technique of contractors, was introduced for the first time in the field of artificial intelligence in the 1970s [36], and it becomes so used in this field.

2.8.2 Definitions [49]

Let us suppose that we have n_x variables $x_i \in \mathbb{R}$, $i \in 1, .., n_x$, linked by n_f constraints in the form [49]

$$f_j(x_1, .., x_{n_x}) = 0, j \in 1, .., n_f. \tag{2.52}$$

Each variable x_i belongs to a given domain $\mathbb{X}_i$. These domains, for simplicity, will be considered as intervals denoted by $[x_i]$. Define the vector X as:

$$X = (x_1, .., x_{n_x})^T \tag{2.53}$$

and the prior domain for X is a box as:

$$[X] = [x_1] \times .. \times [x_{n_x}] \tag{2.54}$$

Contractors	Based on
$C_{GE}\,(Ap-b=0,[A],[p],[b])$	Gauss elimination
$C_{GS}\,(Ap-b=0,[A],[p],[b])$	Gauss-Siedel algorithm
$C_K\,(f(x)=0,[x])$	Krawczyk method
$C_N\,(f(x)=0,[x])$	Newton contractor
$C_{\downarrow\uparrow}\,(f(x)=0,[x])$	forward-backward propagation

Table 2.3: Some of the contractors [49]

The function whose coordinate functions are the f_js, is called f. Eq. 2.52 can be written in vector form as: $f(x)=0$. This corresponds to a *constraint satisfaction problem* (CSP) H [49], which can be formulated as:

$$H:(f(x)=0,x\in[x]) \tag{2.55}$$

The solution set of H is defined as:

$$\mathbb{S}=\{x\in[x]|f(x)=0\} \tag{2.56}$$

Contracting H means replacing $[x]$ by a smaller domain $[x']$ such that the solution set remains unchanged, i.e. $\mathbb{S}\subset[x']\subset[x]$. There exists an optimal contraction of H [49], which corresponds to replacing $[x]$ by the smallest box contains $\mathbb{S}$. A contractor for H is any operator that can be used to contract it.

2.8.3 Basic contractors

A contractor C is an operator used to contract the initial domain of the **CSP**, and to provide a new box.

Many contractors do exist; Krawczyk contractor, Newton contractor, intervalization of Gauss elimination, Gauss-Seidel contractor,... (cf. Table 2.8.3). These contractors work in an efficient way on specific classes of problems only. In our work, we choose the use of the Forward-Backward propagation technique.

First, we present the notion of finite subsolvers to build contractors by intervalization. Then, we present contractors obtained by intervalization of fixed-point methods. The forward-backward contractor is a contractor based on constraint propagation and it is presented later.

In this part, we give the definitions of some of the basic contractors. These definitions are adapted from Luc Jaulin's book [49].

Finite subsolvers

A finite subsolver of the CSP $H : (f(x) = 0, x \in [x])$ ia finite algorithm to compute the values of some variables x_i when others are known x_j.

The vector $u = (u_1, \ldots, u_{n_u})^T$ is a subvector of $x = (x_1, \ldots, x_{n_x})^T$ (with $n_x = Cardinal(x)$), if there exists a subset I of $\{1, \ldots, n_x\}$ of cardinal n_u, such that $n_u < n_x$. I is then called the index set of u, and we shall write $u = x_I$.

Consider $I = (i_1, \ldots, i_{n_u})$ and $J = (j_1, \ldots, j_{n_v})$, two index sets such $I \cap J = \emptyset$, and two subvectors $u = x_I$ and $v = x_J$ of the same vector x. A finite subsolver associated with H is a finite set-valued algorithm $\phi : u \to \phi(u)$ such that the following implication holds true:

$$f(x) = 0 \Rightarrow v \in \phi(u) \tag{2.57}$$

The components of u are called the inputs of ϕ and those of v are called its outputs. In Figure 2.1, the inputs x_1, x_2, x_3 and x_4, and the outputs are x_8 and x_9.

Example: Suppose the following CSP:

$$H : \begin{cases} x_1 x_2 - x_3 = 0 \\ x_2 - sin(x_4) = 0 \\ [x_1] = [x_2] = [-\infty, 0], [x_3] = [x_4] = \mathbb{R} \end{cases} \tag{2.58}$$

We can obtain many subsolvers of *H*, five of them are:

$$\begin{cases} \phi_1(\text{ in: } x_1, x_2; \text{ out: } x_3) \to x_3 = x_1 x_2, \\ \phi_2(\text{ in: } x_1, x_3; \text{ out: } x_2) \to x_2 = x_3/x_1 \text{ if } x_1 \neq 0, \mathbb{R} \text{ otherwise}, \\ \phi_3(\text{ in: } x_4; \text{ out: } x_2) \to x_2 = sin(x_4), \\ \phi_4(\text{ in: } x_1, x_3, x_4; \text{ out: } x_2) \to x_2 = \phi_2(x_1, x_3) \cap \phi_3(x_4), \\ \phi_5(\text{ in: } x_3, x_4; \text{ out: } x_1, x_2) \to x_2 = sin(x_4), x_1 = x_3/x_2 \text{ if } x_2 \neq 0, \mathbb{R} \text{ otherwise}. \end{cases} \tag{2.59}$$

Intervalization of finite subsolvers- Gauss elimination

An important class of CSPs for which intervalization of finite subsolvers can be employed is that of *square linear systems of interval equations*. The problem is to compute a box

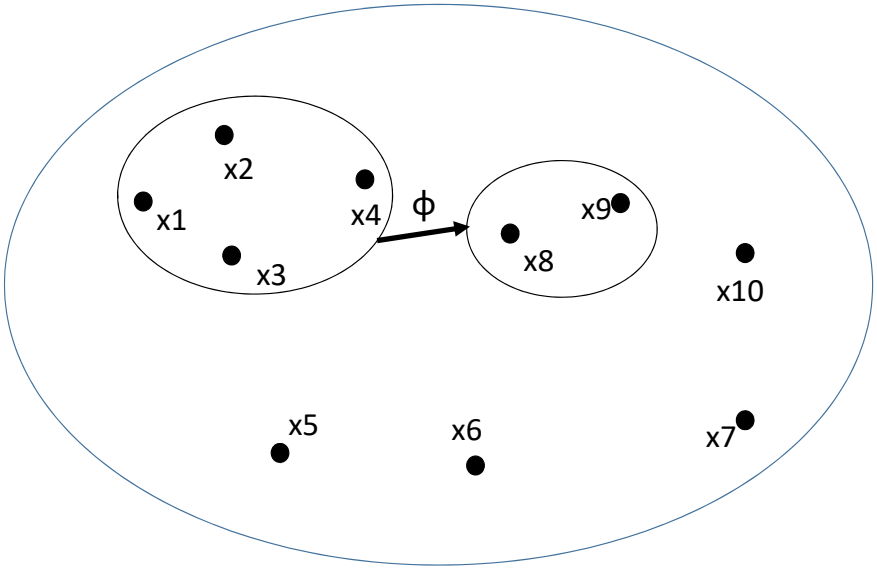


Figure 2.1: Subsolver computing x_8 and x_9 when x_1, x_2, x_3 and x_4 are known

containing the solution set of the CSP.

$$H : \begin{cases} A \in [A], b \in [b], p \in [p], \\ Ap - b = 0 \end{cases} \tag{2.60}$$

with

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n_p} \\ \vdots & \ddots & \vdots \\ a_{n_p1} & \dots & a_{n_pn_p} \end{bmatrix}, b = \begin{bmatrix} b_1 \\ \vdots \\ b_{n_p} \end{bmatrix} \tag{2.61}$$

The variables of H form the vector

$$\begin{cases} x = \left(a_{11}, \dots, a_{n_pn_p}, p_1, \dots, p_{n_p}, b_1, \dots, b_{n_p}\right)^T, \\ C_{GE}\left([A], [b], [p]\right) \rightarrow \left([A], [b], [p] \cap [\phi]([A], [p], [b])\right) \end{cases} \tag{2.62}$$

Fixed-points method: Gauss-Seidel contractor

A fixed-point subsolver for the CSP H:$(f(x) = 0, x \in [x])$ is an algorithm ψ such that

$$f(x) = 0 \Leftrightarrow x = \psi(x) \tag{2.63}$$

A contractor for H is obtained by replacing $[x]$ in H by:

$$[x] \cap [\psi]([x]) \tag{2.64}$$

This contractor is called the fixed-point contractor associated with ψ.
There are many fixed-point contractors, mainly, the interval Gauss-Seidel, Krawczyk and interval Newton contractors. In this part, we present the Gauss-Seidel Contractor.

Consider again the CSP:

$$H : \begin{cases} A \in [A], b \in [b], p \in [p], \\ Ap - b = 0 \end{cases} \tag{2.65}$$

Where the matrix *A* is assumed to be square. We can decompose *A* into the sum of a matrix with zeros on its diagonal and a diagonal matrix:

$$A = diag(A) + extdiag(A) \tag{2.66}$$

Thus, $Ap - b = 0$ will be equivalent to:

$$diag(A)p + extdiag(A)p = b \tag{2.67}$$

Provided that $diag(A)$ is invertible (i.e., *A* has no entry on its diagonal), eq.2.67 can be rewritten as:

$$p = (diag(A))^{-1}(b - extdiag(A)p) \tag{2.68}$$

A fixed-point subsolver for *H* is:

$$\psi \begin{bmatrix} A \\ b \\ p \end{bmatrix} = \begin{bmatrix} A \\ b \\ (diag(A))^{-1}(b - extdiag(A)p) \end{bmatrix} \tag{2.69}$$

An inclusion function for ψ is:

$$[\psi] \begin{bmatrix} [A] \\ [b] \\ [p] \end{bmatrix} = \begin{bmatrix} [A] \\ [b] \\ (diag([A]))^{-1}([b] - extdiag([A])[p]) \end{bmatrix} \tag{2.70}$$

The contractor C_{GS} is given by:

$$C_{GS} : \begin{bmatrix} [A] \\ [b] \\ [p] \end{bmatrix} \rightarrow \begin{bmatrix} [A] \\ [b] \\ [p] \cap (diag([A]))^{-1}([b] - extdiag([A])[p]) \end{bmatrix} \tag{2.71}$$

We should mention that C_{GS} and C_{GE} are efficient when $[A]$ is close to identity matrix.

k	$[p_1](k)$	$[p_2](k)$	$[p_3](k)$
0	$[-10, 10]$	$[-10, 10]$	[-10,10]
1	[-8,9.75]	[-5.40001, 5.00001]	[-9.5,10]
2	[-6.85001, 8.28751]	[-5.17501,4.97501]	[-6.39001,10]
5	[-5.66909,5.24052]	[-3.03602, 4.03031]	[-4.65079,7.84124]
20	[-2.48786, 2.03998]	[-0.994123,2.00077]	[-0.775045,4.29151]

Table 2.4: Iteration of the C_{GS} contractor

Example: Consider the following situation:

$$\begin{cases} [A] = \begin{bmatrix} [4,5] & [-1,1] & [1.5,2.5] \\ [-0.5,0.5] & [-7,-5] & [1,2] \\ [-1.5,-0.5] & [-0.7,-0.5] & [2,3] \end{bmatrix} \\ [b] = \begin{bmatrix} [3,4] \\ [0,2] \\ [3,4] \end{bmatrix} \\ [p] = \begin{bmatrix} [10,10] \\ [10,10] \\ [10,10] \end{bmatrix} \end{cases} \tag{2.72}$$

We get:

$$(diag([A]))^{-1} = \begin{bmatrix} [0.2,0.25] & [0,0] & [0,0] \\ [0,0] & [-0.2,-0.1429] & [0,0] \\ [0,0] & [0,0] & [0.3333,0.5] \end{bmatrix} \tag{2.73}$$

and

$$extdiag([A]) = \begin{bmatrix} [0,0] & [-1,1] & [1.5,2.5] \\ [-0.5,0.5] & [0,0] & [1,2] \\ [-1.5,-0.5] & [-0.7,-0.5] & [0,0] \end{bmatrix} \tag{2.74}$$

The contractor $C_{GS} : ([A],[b],[p])$ yields to:

$$[p] = \begin{bmatrix} [-8,9.75] \\ [-5.4001,5.0001] \\ [-9.5,10] \end{bmatrix} \tag{2.75}$$

When we iterates the contraction using C_{GS}, we obtain in Table 2.4. For k iterations, we obtain the box $[p](k)$ for $[p]$.

Fixed-points method: Krawczyk contractor

Consider the CSP $H : (f(x) = 0, x \in [x])$, where $n_f = n_x$ and **f** is assumed to be differentiable. For any invertible matrix *M*, $\mathbf{f}(x) = 0 \Leftrightarrow x - M\mathbf{f}(x) = x$, the function $\psi(x) = x - M\mathbf{f}(x)$ is a fixed point subsolver for *H*. The centered inclusion function for ψ is

$$[\psi]([x]) = \psi(x_0) + [J_\psi]([x]) * ([x] - x_0) \tag{2.76}$$

where $[J_\psi]$ is an inclusion function for the Jacobian matrix of ψ and $x_0 = mid([x])$.

We obtain the Krawczyk contractor:

$$C_K : [x] \to [x] \cap (\psi(x_0)) + [J_\psi([x])] * ([x] - x_0) \tag{2.77}$$

By replacing $\psi(x)$ with $x - M\mathbf{f}(x)$ in 2.77, we get:

$$C_K : [x] \to [x] \cap (x_0 - M\mathbf{f}(x_0)) + (I - M[J_\mathbf{f}]([x])) * ([x] - x_0) \tag{2.78}$$

Where *I* is the identity matrix and $[J_\mathbf{f}]$ is an inclusion function for the Jacobian matrix of **f**. Usually, we take the matrix **M** as the inverse $J_\mathbf{f}^{-1}(x_0)$ of the Jacobian matrix of **f**, computed at x_0.
Recall the definition of the Jacobian matrix is as follows:
Suppose the function *f* with

$$\mathbf{f:} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \to \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_m(x_1, \ldots, x_n) \end{pmatrix}$$

The Jacobian matrix of the function *f* is defined as:

$$J_\mathbf{f} = \begin{pmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \ldots & \frac{\partial \mathbf{f}}{\partial x_n} \end{pmatrix} \tag{2.79}$$

$$J_\mathbf{f} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \tag{2.80}$$

The inverse of the Jacobian matrix is defined as $J_\mathbf{f}^{-1}$.

Example: Consider the CSP $H : (f(x) = 0, x \in [x])$, described by:

$$H : \begin{cases} f_1(x_1, x_2) = x_1^2 - 4x_2 \\ f_2(x_1, x_2) = x_2^2 - 2x_1 + 4x_2 \\ [x] = [-0.1, 0.1] \times [-0.1, 0.3] \end{cases}$$

The Jacobian matrix for **f** is:

$$J_f = \begin{bmatrix} 2x_1 & -4 \\ -2 & 2x_2 + 4 \end{bmatrix}$$

the inverse of the Jacobian matrix is:

$$J_f^{-1}(x_1, x_2) = \frac{1}{det(J_f)} \begin{bmatrix} J_{f_{22}} & -J_{f_{12}} \\ -J_{f_{21}} & J_{f_{11}} \end{bmatrix}$$

with $det(J_f) = J_{f_{11}} \times J_{f_{22}} - J_{f_{21}} \times J_{f_{12}}$

$$J_f^{-1}(x_1, x_2) = \frac{1}{(4x_1x_2 + 8x_1 - 8)} \begin{bmatrix} 2x_2 + 4 & 4 \\ 2 & 2x_1 \end{bmatrix}$$

M is as follows:

$$M = J_f^{-1}(0, 0.1) = \begin{bmatrix} -0.525 & -0.5 \\ -0.25 & 0 \end{bmatrix}$$

And the Krawczyk contractor yields to:

$$\begin{cases} C_K([x]) = \begin{bmatrix} [-0.0555, 0.005] \\ [-0.005, 0.005] \end{bmatrix} \\ C_K(C_k([x])) = \begin{bmatrix} [-0.00258, 0.00255] \\ [-0.00128, 0.00127] \end{bmatrix} \\ C_K(C_k(C_k([x]))) = \begin{bmatrix} [-0.00000818, 0.00000817] \\ [-0.00000329, 0.00000329] \end{bmatrix} \end{cases}$$

We keep on contracting until the solution converges.

Fixed-points method: Newton contractor

Consider the CSP $H : (f(x) = 0, x \in [x])$, with $n_f = n_x$. The fixed-point subsolver is given by $\psi(x) = x - M\mathbf{f}(x)$, if **f** is affine ($\mathbf{f} = Ax + b$), then we can write down $\psi(x) = x - M(Ax + b)$. With $M = A^{-1}$, the sequence $x_{k+1} = \psi(x_k)$ converges to the solution $x^* = -A^{-1}b$. If **f** is non-linear but differentiable, it can be approximated by its first-order

Taylor expansion to get an approximate fixed-point subsolver $\psi(x) = x - J_{\mathbf{f}}^{-1}(x) * \mathbf{f}(x)$.

The inclusion function is:

$$[\psi]([x]) = [x] - [J_{\mathbf{f}}]^{-1}([x]) * [\mathbf{f}]([x]) \tag{2.81}$$

which leads to the Newton contractor,

$$C_N : [x] \to [x] \cap \left([x] - [J_{\mathbf{f}}]^{-1}([x]) * [\mathbf{f}]([x])\right) \tag{2.82}$$

Forward-backward propagation

As we mentioned before, several contractors do exist, each one of them works in a different way and it is efficient only for specific CSPs and for certain cases.

One popular contractor applied to contracts intervals, that will be used in our approach, is the "Forward-backward propagation (FBP) contractor". This technique is known for its simplicity and ease, it is also more general than the other contractors since it works on all type of systems. It also gives guaranteed results which means that during the contractions we will always get an interval that belongs to the initial interval.

For all these reasons we chose to use the "Forward-backward" propagation technique to help us to contract the intervals in the aim to compute the imprecise availability when *t* tends to infinity as given in the system of equation $\Pi.Q = 0$, in the case of imprecise data.

In the next subsection, we detail the technique of forward-backward propagation, which it will be used later.

2.8.4 Methodology of the Forward-Backward contractor [49]

Forward-backward (FBP) contractor $C_{\downarrow\uparrow}$ is a classical algorithm in constraint programming for contracting. It is based on constraint propagation [49].

Using this contractor makes it possible to contract the domains of the CSP

$$H : (f(x) = 0, x \in [x]) \tag{2.83}$$

by taking into account each one of the n_f constraints apart, say $f_i\left(x_1, \ldots, x_{n_x}\right)$. In this case, n_f is not necessarily equal to n_x.

The algorithm works in two steps. The *forward* step applies interval arithmetic to each operator of the function $y = f(x)$, from the variable's domain $([x])$ up to the function's domain $([y])$, this step considers the direct forms of the equations.

The *backward* step sets the interval associated to the new function's domain $[y]$ to $[0, 0]$ (imposes constraint satisfaction, since we are solving $f(x) = 0$) and, then, applies backward arithmetic from the function's domain to the variable's domain, which means using the inverse of the functions that appear in the equations f(x). The following example explains the procedure of the FBP technique.

Example: Consider the constraint $y = -5x_1 + 2x_2 = 0$ and the initial box-domain $[x] = [1, 4] \times [-3, 7]$.

This constraint can be decomposed as shown in eq. 2.84 into three primitive constraints (i.e., constraints involving a single operator such as $(+, -, *, /)$ or a single function) by introducing two intermediate variables a_1 and a_2 defined as: $a_1 = -5.x_1$ *and* $a_2 = 2.x_2$. Initial domains for these variables are determined as follows:

$$\begin{cases} [a_1] := -5[x_1] = -5 \times [1, 4] = [-20, -5] \\ [a_2] := 2[x_2] = 2 \times [-3, 7] = [-6, 14] \\ [y] := [a_1] + [a_2] = [-20, -5] + [-6, 14] = [-26, 9] \end{cases} \tag{2.84}$$

and this step is called the "forward propagation". A method for contracting H with respect to the constraint $f(x) = 5x_1 + 2x_2 = 0$ is to contract each of the primitive constraints in 2.84 until the contractors become inefficient, which the solution converges.
For this example: Since $f(x) = 0$, the domain for y should be taken equal to 0, we can add the step:

$$[y] := [y] \cap 0 \tag{2.85}$$

If $[y]$ as computed in eq. 2.85 turns out to be empty, then the CSP has no solution. Else, $[y]$ is replaced by 0, which is the case in this example. After, a backward propagation is performed, updating the domains associated with all the variables to get:

$$\begin{cases} [a_1] := ([y] - [a_2]) \cap [a_1] = ([0, 0] - [-6, 14]) \cap [-20, -5] \Longrightarrow [a_1] = [-14, -5] \\ [a_2] := ([y] - [a_1]) \cap [a_2] = ([0, 0] - [-20, -5]) \cap [-6, 14] \Longrightarrow [a_2] = [5, 14] \\ [x_1] := ([a_1]/-5) \cap [x_1] = ([-14, -5]/-5) \cap [1, 4] \Longrightarrow [x_1] = [1, 14/5] \\ [x_2] := ([a_2]/2) \cap [x_2] = ([5, 14]/2) \cap [-3, 7] \Longrightarrow [x_2] = [5/2, 7]. \end{cases} \tag{2.86}$$

Step 1: Forward	$a_1 := -5x_1$
Step 2: Forward	$a_2 := 2x_2$
Step 3: Forward	$y := a_1 + a_2$
Step 4	$[y] := [y] \cap 0$
Step 5: Backward	$[a_1] := ([y] - [a_2]) \cap [a_1]$
Step 6: Backward	$[a_2] := ([y] - [a_1]) \cap [a_2]$
Step 7: Backward	$[x_1] := ([a_1]/-5) \cap [x_1]$
Step 8: Backward	$[x_2] := ([a_2]/2) \cap [x_2]$

Table 2.5: Steps of $C_{\downarrow\uparrow}$

And we obtain the new box :

$$[x](1) = [1, 14/15] \times [5/2, 7] \tag{2.87}$$

which is the result of the first FBP contraction. Iterating this procedure, the resulting sequence of boxes $[x](k)$ converges towards the smallest possible domain, after which the domains no longer change following another iteration of FBP. Table 2.5 shows the corresponding steps of the forward-backward contractor applied on this example.

Here another example is token from Luc Jaulin's book [49]. Consider the CSP:

$$H : \begin{cases} x_1 + 2x_2 - x_3 = 0 \\ x_1 - x_2 - x_4 = 0 \\ [x] \in [-10, 10] \times [10, 10] \times [-1, 1] \times [-1, 1] \end{cases}$$

Using $C_{\downarrow\uparrow}$, the constraint $x_1 + 2x_2 - x_3 = 0$ gives:

$$\begin{cases} [x_1] = [-10, 10] \\ [x_2] = [-\frac{11}{2}, \frac{11}{2}] \\ [x_3] = [-1, 1] \end{cases}$$

and the constraint $x_1 - x_2 - x_4 = 0$ gives:

$$\begin{cases} [x_1] = [-\frac{13}{2}, \frac{13}{2}] \\ [x_2] = [-\frac{11}{2}, \frac{11}{2}] \\ [x_4] = [-1, 1] \end{cases}$$

When iterating the steps of the contractor, a sequence of boxes $[x](k)$ converge to the smallest possible domain, see Table 2.6. The new box is:

$$[x] = [-3, 3] \times [-2, 2] \times [-1, 1] \times [-1, 1]$$

k	$[x_1](k)$	$[x_2](k)$	$[x_3](k)$	$[x_4](k)$
0	[-10,10]	[-10,10]	[-1,1]	[-1,1]
1	[-6.5,6.5]	[-5.5,5.5]	[-1,1]	[-1,1]
2	[-4.75,4.75]	[-3.75,3.75]	[-1,1]	[-1,1]
5	[-3.2,3.2]	[-2.2,2.2]	[-1,1]	[-1,1]
10	[-3.006,3.006]	[-2.006,2.006]	[-1,1]	[-1,1]
∞	[-3,3]	[-2,2]	[-1,1]	[-1,1]

Table 2.6: Iterations of $C_{\downarrow\uparrow}$

2.9 Conclusion

This chapter has presented the main idea of interval analysis. We showed all the basic definitions of interval operators. We introduced the idea of contractors and presented some of the main contractors. The Gauss-Siedel method is useful in some cases. However, the use of this method demands some conditions (for example all the diagonal elements of the matrix must be different than zero). Also, it is a slow algorithm (it requires many iterations), each iteration can require a lot of computing time, depending on the case. The Krawczyk and Newton contractors are very useful when we have nonlinear differential functions and matrices, in the aim to apply contractions on the two of them. In our case, we have a system of linear equations where we do not want to contract the matrix so we can use another contractor that is simple to apply. The Forward-Backward contractor is the most used contractor. It is a general contractor used on linear equations and nonlinear equations. It is very simple to apply; the operations are based on simple arithmetic calculations to facilitate the contractions. The Forward-Backward contractor gives always a guaranteed result where it includes the real exact values. The Forward-Backward contractor and the Gauss-Siedel contractor yield to close results, however the number operations, the number of contractions and the computing time vary. In our case, the equations are linear where we do not want to contract the matrix, therefore we use the simplest contractor among all of them which is the Forward-Backward contractor.

In this chapter, we developed the "forward-backward" contractor since we will use it later in our methodology.

The following chapter is devoted to present the difference between a binary system and a multi-states system (MSS). Then, we present the state of art on dependability in particularly about the works that have been done with complex MSS. Also, we show all the works that have been done in this subject.

Chapter 3

Dependability state of art

3.1 Introduction

In dependability studies, we use a set of tools and methods that allow, in all levels of a system's life, to ensure that the system performs or accomplishes the missions for which it was designed, and in conditions of reliability, maintainability, availability and security predefined. Generally, these studies consist in analyzing the effects of failures, malfunctions, errors of the studied system. It should be noted that a lot of works have advanced research and have been developed within the framework of dependability. In this, chapter we cite some of the related works.

Systems are often multi-states. During operation, the lack of data leads to uncertainties. For that, in this thesis, we are interested in multi-states systems (MSS) imprecise availability assessment. However, before passing to the part of MSSs which is our main subject, in section 2, we present some of the related works in dependability for binary systems.

Later in section 3, we present some of the related works in dependability when dealing with MSS.

Among all the methods of representing the system, we chose the Markov model method to help us to calculate the availability. Section 4 presents the explanations and the related works in this topic.

Section 5, is about uncertainties in dependability. Therefore, this section concerns the works on imprecise availability computing.
To end up with conclusion and the problematic of this work.

3.2 Binary systems methods

Before discussing MSSs, we will present the different methods and techniques used to estimate the availability (or reliability) of binary systems with two operating states. Which are a particular case of MSSs whose elements do not work with degraded performance levels between the two extreme states "perfect working" and "total failure". Of course, there are more effective, more complex, more precise, more general or faster methods than others, but each of these methods has its different advantages and disadvantages, and for each type of structure one may appear more efficient than the other.

To confirm these availability assessment models, we will determine the most appropriate technique for each type of structure. There are several works in the literature that present in general the different methods and techniques that have been developed to deal with the problems of the reliability of binary systems, then these methodologies have been evolved and adapted to solve a wide variety of problems.

3.2.1 States enumeration technique

When the system is composed of a small number of components, its reliability can be calculated directly by listing all the possible configurations. The reliability of the total system is found by doing the sum over all the probabilities that present the system when it is working.

Consider the stochastic graph **G** defined by the $\mathbf{G} = (V, U)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, and $U = \{u_1, u_2, \ldots, u_n\}$ is the set of arcs that represent the components.

An arc u of the set U is defined by a pair of nodes. The arc $u = (a, b)$, means that the arc goes from a to b. It is also said that a is the initial extremity and b is the final extremity.

In the graph in Figure 3.1, the nodes are $V = \{1, 2, 3, 4, 5, 6\}$ and the arcs are $U = \{(1, 2), (1, 3), (3, 4), (3, 6), (4, 1), (5, 2), (5, 6), (6, 3)\}$.

We note $d^+(v)$ the **external degree** of the node v, i. e. the number of arcs having v as the initial extremity.
We note $d^-(v)$ the **inner degree** of the node v, i. e. the number of arcs that have v as the final extremity.

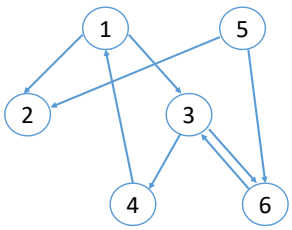


Figure 3.1: Oriented graph

The **degree of the node** v is:

$$d(v) = d^+(v) + d^-(v) \tag{3.1}$$

In the graph in Figure 3.1, we have the following degrees:

$$\begin{cases} d^+(1) = 2, d^+(2) = 0, d^+(3) = 2, d^+(4) = 1, d^+(5) = 2, d^+(6) = 1 \\ d^-(1) = 1, d^-(2) = 2, d^-(3) = 2, d^-(4) = 1, d^-(5) = 0, d^-(6) = 2 \end{cases}$$

A **path** leading from the top a at the top z is a sequence of nodes and arcs (v, u), beginning and ending with a node, such that each arc is bounded on the left by its original node and on the right by its destination node.

On the graph in Figure 3.1, we can take for example the paths: $(1, (1, 2), 2)$ and $(5, (5, 6), 6, (6, 3), 3, (3, 4), 4, (4, 1), 1, (1, 2), 2)$.

A **circuit** is a path where its starting and ending points are the same. The circuit that can be drawn from the graph in Figure 3.1, is for example $(1, (1, 3), 3, (3, 4), 4, (4, 1), 1)$.

Let $\mathbf{G}' = (V', U')$ a sub-graph of $\mathbf{G} = (V, U)$ with $U' \subseteq U$. Let Γ be the set of all the sub-graphs $\mathbf{G}'$ of $\mathbf{G}$ such that the end points S and T are included in V'. The reliability of the system can be written as follows:

$$\begin{cases} R(\mathbf{G}) = \sum_{\mathbf{G'}_i \in \Gamma} Pr(\mathbf{G'}_i) = 1 - \sum_{\mathbf{G'}_i \notin \Gamma} Pr(\mathbf{G'}_i) \\ P(\mathbf{G'}_i) = \prod_{j \in \mathbf{G'}_i} p_j \cdot \prod_{j \notin \mathbf{G'}_i} (1 - p_j) \end{cases} \tag{3.2}$$

with $j \in U$.

This method seems simple and efficient, however the cost of computation in terms of time increases exponentially with the extension of the graph size (number of components) [68].

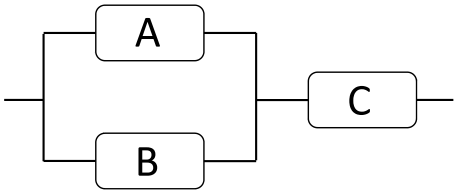


Figure 3.2: Block diagram of the system

3.2.2 Technique of the Boolean Truth Table

The Boolean truth table approach is a simple method, it is based on listing all the possible combinations of the states of the system's components. It requires knowledge of the elements that lead to system failure and the link between the system's components.

The truth table contains n entries and 2^n lines (2^n possible combinations), such that n is the number of system's components. Each column represents an element of the system and the lines represent the different states of these elements; a "1" means that the component is working and a "0" means that the component isn't working. Each line is tested independently to define the status of the complete system (example Figure 3.2 and Table 3.1).

The probability of each combination is calculated by multiplying the probabilities of the component states. Then, the reliability of the system is obtained by summing all the probabilities of the combinations that lead to the system functioning.

This method is practical only for small systems. Otherwise, the approach becomes very limited, in fact the truth table of such a system is complex and long, as it is not always easy to predict the state of the complete system from the states of its components.
With:

$$R_s = P_4 + P_6 + P_8 = 1 - (P_1 + P_2 + P_3 + P_5 + P_7)$$

3.2.3 Inclusion-exclusion method

This method requires the enumeration of all minimum paths or minimum cuts of the studied system.

A	B	C	System	Probability
0	0	0	0	$P_1 = (1 - P_A).(1 - P_B).(1 - P_C)$
0	0	1	0	$P_2 = (1 - P_A).(1 - P_B).P_C$
0	1	0	0	$P_3 = (1 - P_A).P_B.(1 - P_C)$
0	1	1	1	$P_4 = (1 - P_A).P_B.P_C$
1	0	0	0	$P_5 = P_A.(1 - P_B).(1 - P_C)$
1	0	1	1	$P_6 = P_A.(1 - P_B).P_C$
1	1	0	0	$P_7 = P_A.P_B.(1 - P_C)$
1	1	1	1	$P_8 = P_A.P_B.P_C$

Table 3.1: Boolean truth table

The two concepts "**minimum path**" and "**minimum cut**" has been presented in section 7 of chapter 1. The computing of the reliability of the system is based on on Poincaré's formula. Through this formula, the reliability of the system is expressed by the probability of union of all the minimum cuts or all the minimum paths (the choice of using minimum paths or minimum cuts depends on the total number of the minimum paths and the total number of the minimum cuts, we choose the lowest one) and their intersection probabilities. In general, calculations can be made using either the minimum path technique or the minimum cut technique. The adoption of one of these two approaches is based on the number of minimum paths and minimum cuts that can be extracted from the system, when the number is lower, the calculation is more simple.

Let us consider the structure of Figure 3.2, we can detect two minimum paths $C_1 = AC$ and $C_2 = BC$ and two minimum cuts $D_1 = \overline{AB}$ and $D_2 = \overline{C}$. The well-functioning of at least one path implies the functioning of the system, therefore the reliability assessment using the minimum paths is calculated as follows:

$$\begin{cases} R_s = P(C_1 \cup C_2) = P(C_1) + P(C_2) - P(C_1 \cap C_2) \\ R_s = P_A.P_C + P_B.P_C - P_A.P_B.P_C \end{cases} \tag{3.3}$$

From the minimum cuts, we can calculate the probability that the system is nonfunctional:

$$\begin{cases} F_s = 1 - R_s = P(D_1 \cup D_2) = P(D_1) + P(D_2) - P(D_1 \cap D_2) \\ F_s = (1 - P_A).(1 - P_B) + (1 - P_C) - (1 - P_A).(1 - P_B). \end{cases} \tag{3.4}$$

We form from Poincaré's formula for m minimum paths the following formula:

$$P(\bigcup_{i=1}^{m} C_i) = \sum_{i=1}^{m} P(C_i) - \sum_{1\leq i<j\leq m} P(C_i \cap C_j) + \sum_{1\leq i<j<k\leq m} P(C_i \cap C_j \cap C_k) + \ldots + (-1)^{m+1} P(C_1 \cap \ldots \cap C_m) \tag{3.5}$$

The good thing about this method that it can be applied on a wide range of problems regardless of the type of the system. However, the calculation of the reliability using the formula will not be simple when it comes to a large system.

3.3 MSS methods

In this section, we will discuss the different approaches and techniques for computing the reliability of MSSs, where the operating states present in degradation.

An MSS is the general case of a system and the binary system is a particular case of the MSS. An MSS is a system where the components and the system could have different degraded states from "perfect working" to " total failure".

As we have seen, there are many techniques that have been developed to model and to calculate the reliability of binary systems. However, a lot of these methods cannot be applied to MSSs. The problem is that when having this type of system, the reliability calculation become hard. Since the size of the system where the elements of the system can have many different states, this will cost in terms of time to solve the problem.

Since 1998, the two researchers G. Levitin and A. Lisnianski have published several works in this perspective, their works were dedicated to the analysis, to the calculation and the optimization of the availability of an MSS. They have improved some of the approaches that simplify the systems by significantly reducing calculation time. Generally, methods for assessing the reliability of MSSs are based on four different approaches: an extension of **Boolean models** [71], the **stochastic process** (mainly Markovian and semi-Markovian) [27], the approach of the **Universal Generating Function** (UGF) [63] and **Monte-Carlo** simulation technique [13]. With regard to this subject, in their book published in 2003 [64, 71], the two authors G. Levitin and A. Lisnianski presented several models to assess the reliability of MSSs.

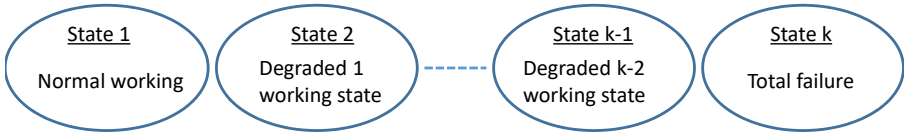


Figure 3.3: Functional diagram of a multi-states component

In this section, first we will detail the concept of the MSS. Later, we will present the four methods mentioned before and we will show some of the related works.

3.3.1 Description of a MSS

A MSS is a system that can operate with degraded performance levels between normal functioning and total failure. A component is an entity of the system that cannot be broken down into other components. This entity itself can have several operating states. Therefore, an MSS can be made up of binary elements or multi-states elements. Figure 3.3 shows a functional diagram of a component having k possible states.

In order to analyze a MSS, it is necessary to know the characteristics of its components. Each element i of a system can have k_i different states, each state corresponds to a performance level. The different performance levels are represented by the set:

$$\boldsymbol{G}_i = \{g_{i1}, g_{i2}, \ldots, g_{ik_i}\} \tag{3.6}$$

At time t, the performance of the element i noted by $G_i(t)$ is a random variable that takes its values from the set $\boldsymbol{G}_i$.

The associated probabilities to the states g_{ij} of the i with $j \in \{1, 2, \ldots, k_i\}$ are represented by the set:

$$\boldsymbol{P}_i = \{p_{i1}, p_{i2}, \ldots, p_{ik_i}\} \tag{3.7}$$

with

$$p_{ij} = \text{Probability}\,\{G_i(t) = g_{ij}\} \tag{3.8}$$

The performance level of a system with n multi-states components is defined from the performance levels of its components. The entire system can have k different states. The

performance level $G(t)$ of the system is a random variable that takes its values from the set:

$$\boldsymbol{G} = \{g_1, g_2, \ldots, g_k\} \tag{3.9}$$

The space of all the possible combinations of the performance levels of the n components of the system is:

$$L^n = \{g_{11}, g_{12}, \ldots, g_{1k_1}\} \times \{g_{21}, g_{22}, \ldots, g_{2k_2}\} \times \ldots \times \{g_{n1}, g_{n2}, \ldots, g_{nk_n}\} \tag{3.10}$$

The structure function associates each combination in L^n, to the performance level of the corresponding total system:

$$\begin{cases} \phi : L^n \rightarrow \boldsymbol{G} \\ \phi\left(G_1(t), G_2(t), \ldots, G_n(t)\right) = G_t \end{cases} \tag{3.11}$$

The general model of a MSS can be defined as follows:

$$\begin{cases} g_i, p_i(t), 1 \leq i \geq n \\ \phi\left(G_1(t), G_2(t), \ldots, G_n(t)\right) \end{cases} \tag{3.12}$$

With g_i is the state of component i and p_i is the probability that element i is in the state g_i.

The required performance limit rated $W(t)$ (or required demand) determines the flow that must be generated by the system at the output node. From this limit, the set of the system states $\boldsymbol{G}$ (see eq.3.9) can be divided into two separate subsets $\boldsymbol{G'} = \{\text{accepted states}\}$ and $\overline{\boldsymbol{G}} = \{\text{unaccepted states}\}$ with $\boldsymbol{G'} \cup \overline{\boldsymbol{G}} = \boldsymbol{G}$. The first set includes the system states that provide a flow greater than or equal to the required demand:

$$\boldsymbol{G'} = \{G(t) \in \boldsymbol{G} / G(t) \geq W(t)\} \tag{3.13}$$

The second set includes the system states that generate a flow less than the required demand:

$$\overline{\boldsymbol{G}} = \{G(t) \in \boldsymbol{G} / G(t) < W(t)\} \tag{3.14}$$

The **availability** of a system is the ability that it is in its working states at a instant t. The availability can be written as follows:

$$A(t) = \sum_{G_i \in G} Pr\left\{G(t) \geq W(t)\right\} = \sum_{G_i \in G'} q_i(t) \tag{3.15}$$

With $q_i(t)$ is the probability of having the combination of component performances that ensures the level of performance $G_i(t)$. If we assume that $G_i(t) = \phi\left(g_{1i}(t), g_{2i}(t), \ldots, g_{ni}(t)\right)$ then $q_i(t)$ is:

$$q_i(t) = \prod_{j=1}^{n} p_{ij} \tag{3.16}$$

In the following sections, we will present some of the main methods applied on MSS to calculate its availability.

3.3.2 The Universal Generating Function (UGF)

The principles of the UGF method were introduced by I. Ushakov in 1986 [113], then many scientists (G. Levitin, A. Lisnianski,. . .) proved the capacity of the method [63]. This method is based on simple algebraic operations and determines the performance level and the availability of the overall system from the characteristics of its components.

Mathematical fundamentals

Suppose n independent discrete random variables $X_1, \ldots, X_n$ and suppose that each variable X_i can be presented by the vectors x_i and p_i:

$$\begin{cases} x_i = (x_{i1}, \ldots, x_{ik_i}) \\ p_i = (p_{i1}, \ldots, p_{ik_i}) \text{ with } p_{ij} = Pr\{X_i = x_{ij}\} \text{ with } j = 0, \ldots, k_i \end{cases} \tag{3.17}$$

The evaluation of the performance distribution function of a function $(X_1, \ldots, X_n)$, requires the evaluation of the vector y of all the possible values that this function could take and the vector q of the corresponding probabilities.

If we consider that each variable X_i can have k_i different implementation, then the number of the possible combinations is:

$$K = \prod_{1 \leq i \leq n} k_i \tag{3.18}$$

However, the variables X_i are independent, therefore the probability of having each combination is equal to the product of the probabilities of the realizations of the arguments of the combination. For a combination $C_j = (x_{1j_1}, \ldots, x_{nj_n})$ such that each x_{ij_i} is a value of the vector x_i for $1 \leq i \leq n$, the value of the corresponding function f is the combination probability, we have:

$$\begin{cases} f_j = f(C_j) \\ q_j = \prod_{i=1}^{n} p_{ij_i} \text{ with } p_{ij_i} = Pr\{X_i = x_{ij_i}\} \end{cases} \tag{3.19}$$

Several combinations can have the same value of the function f, then the probability that the function f takes the same value f_h is equal to the sum of the probabilities of the combinations that give this value, in case of independent combinations. Let H be the number of the possible values of f, then the set of combinations producing the same value f_h is:

$$A_h = \{C_j / f_j = f_h, 1 \leq j \leq H\} \tag{3.20}$$

UGF applied on MSSs

Consider a MSS with n multi-states components. Each component i could have k_i possible states, such that G_{ij_i} is the performance level of state j_i of component *i* and p_{ij_i} the probability of being in this state. To evaluate the performance distribution, we write the U-function of component *i* as:

$$u_i(Z) = \sum_{j_i=1}^{k_i} p_{ij_i} Z^{G_{ij_i}} [38] \tag{3.21}$$

To obtain the UGF of m components, we use the the composition operators. Based on simple algebraic operations, we find the $U(Z)$ of a group of components, and the resulted U-function of these m components is:

$$\Omega\left(u_1(Z), u_2(Z), \ldots, u_m(Z)\right) = \sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \cdots \sum_{j_m=1}^{k_m} \left(\prod_{i=0}^{m} p_{ij_i} Z^{w(G_{ij_1}, \ldots, G_{ij_m})} \right) \tag{3.22}$$

where the function $w\left(G_{ij_1}, \ldots, G_{ij_m}\right)$ represents the performance of m components, if they are connected in series it will be there minimum, with:

$$w\left(G_{ij_1}, \ldots, G_{ij_m}\right) = min\left(G_{ij_1}, \ldots, G_{ij_m}\right) \tag{3.23}$$

if they are simultaneous active redundancy the function w will be the sum of their performances, with:

$$w\left(G_{ij_1}, \ldots, G_{ij_m}\right) = \sum_{i=1}^{m} G_{ij_i} \tag{3.24}$$

Let us consider that there is r combinations $(G_1, \ldots, G_m)$ that allow the system to produce the same level G_i. The probability of the occurrence of the jth combination $(G_1, \ldots, G_m)$ which ensures this level, is q_{ij}. Then the probability of having the performance level G_i is:

$$q_i = \sum_{j=1}^{r} q_{ij} \tag{3.25}$$

According to eq. 3.9, the set *G* represents the performance levels that the system could have. Suppose that the cardinal of *G* is the l, then the U-function of the global system takes the form:

$$U(Z) = q_1 Z^{G_1} + q_2 Z^{G_2} + \ldots + q_l Z^{G_l} \tag{3.26}$$

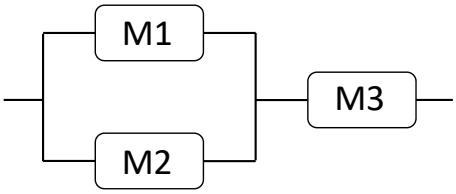


Figure 3.4: System of 3 machines

with $G_i \in G$ such as $G_1 < G_2 < \ldots < G_l$ and $q_i = \sum_{j=1}^{r_i} q_{ij}$ such as $\sum_{i=1}^{l} q_i = 1$.

The availability of the MSS is the probability that the performance $G(t)$ of the system is greater than the required demand $W(t)$. Let $G_k = \{min(G_1, G_2, \ldots, G_l)/G_k \geq W(t)\}$. According to eq. 3.15 and eq. 3.26, the availability can be written:

$$A(t,w) = \sum_{G_i \in G} Pr\left\{G_i(t) \geq W(t)\right\} = \sum_{i=k}^{l} q_i \tag{3.27}$$

This method gives a fast estimation but its application remains limited to systems with simple structure: series, parallel, series-parallel,... This method wouldn't be efficient when the structure of the system is complex.

Numerical application

This example is token from the Master's report of Kaoutar Rhazali [95]. In a production process, there are three machines M_1, M_2 and M_3. First, the product must pass through either M_1 or M_2 and then through M_3. The system is illustrated in Figure 3.4:

Since M_1 and M_2 perform the same task then they are modeled by a parallel connection. M_3 is modeled by a serial connection with M_1 and M_2 because all products leaving them must pass through M_3.

Suppose that M_1 can process a flow of $\phi_1^1 = 1000p/h$ (pieces per hour), when it operates normally. In case of an intermediate failure, the flow is reduced to $\phi_1^2 = 600p/h$. In case of a total failure, the flow becomes zero $\phi_1^3 = 0p/h$. M_2 machine is binary, thus, either it produces a flow of $\phi_2^1 = 500p/h$ or it is in failure mode with $\phi_2^2 = 0p/h$. M_3 produces a flow $\phi_3^1 = 1500p/h$ when it is in normal operation. According to the first failure case, the flow is reduced to $\phi_3^2 = 1000p/h$, according to the second failure case the flow decreases to

M_1		M_2		M_3	
Performance (G)	**Probability (p)**	**Performance (G)**	**Probability (p)**	**Performance (G)**	**Probability (p)**
1	0.8	0.5	0.95	1.5	0.69
0.6	0.15	0	0.05	1	0.2
0	0.05	-	-	0.5	0.1
-	-	-	-	0	0.01

Table 3.2: Characteristics of M_1, M_2 and M_3

$\phi_3^3 = 500p/h$ and when it is in total failure the flow becomes zero $\phi_3^4 = 0p/h$.

It is assumed that the system operates normally only when the output flow is equal to $1000p/h$. Then the required demand is $W = \frac{1000}{1000} = 1$.

M_1 has 3 states, the corresponding performance levels are:

$$\begin{cases} g_{11} = \frac{1000}{1000} = 1; \\ g_{12} = \frac{600}{1000} = 0.6; \\ g_{13} = 0; \end{cases}$$

M_2 has only 2 states, the performance levels are:

$$\begin{cases} g_{21} = \frac{500}{1000} = 0.5; \\ g_{22} = 0; \end{cases}$$

M_3 has 4 operating states, the corresponding performance levels are:

$$\begin{cases} g_{31} = \frac{1500}{1000} = 1.5; \\ g_{32} = \frac{1000}{1000} = 1; \\ g_{33} = \frac{500}{1000} = 0.5; \\ g_{34} = 0; \end{cases}$$

Table 3.2 presents the states of the three components and their probabilities:
The number of combinations of the states is:

$K = \prod_{i=1}^{3} k_i$ with $k_1 = 3$, $k_2 = 2$ and $k_3 = 4$ which gives $K = 3 \times 2 \times 4 = 12$

the system has 12 combinations C_j.

Table 3.3 represents the U-functions of each machine:
The two blocks M_1 and M_2 are connected in parallel:

Machine	U**-function**
M_1	$u_1(Z) = 0.8Z^1 \times 0.15Z^{0.6} \times 0.05Z^0$
M_2	$u_2(Z) = 0.95Z^{0.5} \times 0.05Z^0$
M_3	$u_3(Z) = 0.69Z^{1.5} \times 0.2Z^1 \times 0.1Z^{0.5} \times 0.01Z^0$

Table 3.3: U-functions of the three machines

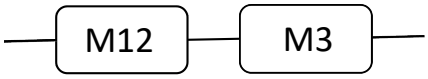


Figure 3.5: Simplified reability block diagram of Figure 3.4

$w(G_{1j_1}, G_{2j_2}) = G_{1j_1} + G_{2j_2}$ with G_{ij_i} the state of j_i of the ith machine ($j_1 \in 1, 2, 3$ and $j_2 \in 1, 2$)

$\Omega(u_1(Z), u_2(Z)) = \Omega\left(0.8Z^1 + 0.15Z^{0.6} + 0.05Z^0, 0.95Z^{0.5} + 0.05Z^0\right) = 0.8{\times}0.95Z^{w(1,0.5)} + 0.8{\times}0.05Z^{w(1,0)} + 0.15{\times}0.95Z^{w(0.6,0.5)} + 0.15{\times}0.05Z^{w(0.6,0)} + 0.05{\times}0.95Z^{w(0,0.5)} + 0.05{\times}0.05Z^{w(0,0)} = 0.76Z^{1.5} + 0.04Z^1 + 0.1425Z^{1.1} + 0.0075Z^{0.6} + 0.0475Z^{0.5} + 0.0025Z^0$.

Thus,

$$u_{12}(Z) = 0.76Z^{1.5} + 0.1425Z^{1.1} + 0.04Z^1 + 0.0075Z^{0.6} + 0.0475Z^{0.5} + 0.0025Z^0$$

If we merge the two blocks M_1 and M_2 into a single block noted M_{12}, then the reliability block diagram will be simplified as Figure 3.5 shows.

The two blocks M_{12} and M_3 are in series:

$$w(G_{12j_{12}}, G_{3j_3}) = min(G_{12j_{12}}, G_{3j_3})$$

with G_{ij_i} the performance level of the state j_i of the machine i, with $j_{12} \in 1, 2, 3, 4, 5, 6$ and $j_3 \in 1, 2, 3, 4$.

$\Omega(u_{12}(Z), u_3(Z)) = \Omega(0.76Z^{1.5} + 0.1425Z^{1.1} + 0.04Z^1 + 0.0075Z^{0.6} + 0.0475Z^{0.5} + 0.0025Z^0, 0.69Z^{1.5} \times 0.2Z^1 \times 0.1Z^{0.5} \times 0.01Z^0)$

When we form the U-function of the system whe obtain:
$U(Z) = 0.012475Z^0 + 0.142025Z^{0.5} + 0.006675Z^{0.6} + 0.2161Z^1 + 0.098325Z^{1.1} + 0.5244Z^{1.5}$

According to the U-function of the system, we can notice that this one can provide three performance levels greater than or equal to the required demand $W = 1$ ($G_4 = 1, G_5 = 1.1$ and $G_6 = 1.5$). The availability is:

$$A(W = 1) = \sum_{i=3}^{6} q_i = 0.8388$$

3.3.3 Inclusion-Exclusion method

When having a system that it can't be broken down into subsystems with simple structures, the use of the UGF method becomes very limited. Indeed, in this case we talk about complex system and there is no function that helps to calculate the U-function of such systems.

In the case of binary systems with complex structures, the "Inclusion-Exclusion" technique is considered to be the most efficient method for computing its availability. In [77, 76, 77] an algorithm has been developed based on the extension of this method to apply on MSSs.

The calculation of reliability based on the minimum paths makes it possible to evaluate the reliability for any type of system. This method seems simple but when it comes to MSSs, the meaning of the term of minimum path changes. Indeed, the minimum path becomes conditioned by the performance level. For each state of the system, we can extract various minimum paths depending on the states of the components.

Formulation of the Problem

In this section, we will develop the "Inclusion-Exclusion" method for computing the availability of an MSS. This method is based on the extension of the minimum path approach applied to MSSs.

The method is applied to any system consisting of n subsystems such as each subsystem s_i can be composed from n_i elements with $i \in \{1, 2, \ldots, n\}$ (cf. Figure 3.6). The components are numbered from 1 to N from left to right and from top to bottom with $N = \sum_{i=1}^{n} n_i$.

Each component of the subsystem i can receive a flow from one or many components of the subsystem $i - 1$, as it can also divide its flow between one or more components of the subsystem $i + 1$.

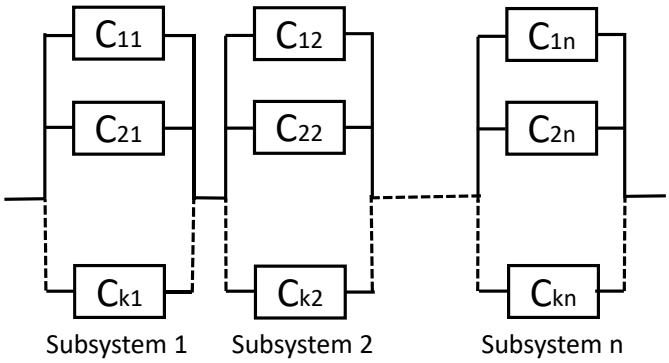


Figure 3.6: Parallel-series system

Let's consider that each component j can have the following performance levels $\{0, 1, \ldots, K_j\}$ with $j \in \{1, 2, \ldots, N\}$. The probability that this component is in the state k is noted by p_{jk} with $k \in \{0, 1, 2, \ldots, K_j\}$. The parameter W represents the required demand. The vector X with N its length, represents the states of the system components, it can take values from $(0, 0, \ldots, 0)$ to $(K_1, K_2, \ldots, K_N)$.

We should know that the structure function ϕ associates to each vector X, a state h of the system.

Principles of the method

X is called the **minimum path vector for a level** h if the image of X using the function ϕ is greater than or equal to h ($\phi(X) \geq h$) and that for any vector $Y < X$, the image of Y by ϕ is strictly less than h ($\phi(Y) < h$).

X is called the **minimum cut vector for a level** h if the image of X using the function ϕ is strictly less than h ($\phi(X) < h$) and that for any vector $Y > X$, the image of Y by ϕ is greater than or equal to h ($\phi(Y) \geq h$).

NB: We say that $Y > X (Y < X)$, if there is one $i \in \{1, 2, ..., N\}$ such that: $y_i > x_i (y_i < x_i)$ and for any $k \neq i$ with $k \in 1, 2, \ldots, N$, we have: $y_k = x_k$.

y_i corresponds to the state of component i, $Y = (y_1, y_2, \ldots, y_N)$.

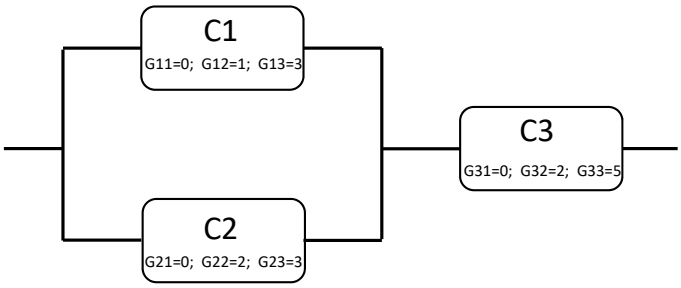


Figure 3.7: Example of minimal cut and minimal path for a level h

Consider a system of three components and suppose that each component can operate at three different levels (cf. Figure 3.7):

- The minimum path vectors:
  - For level $h = 1$: $X = (1, 0, 2)$
  - For level $h = 2$: $X = (3, 0, 2)$ and $X = (0, 2, 2)$
  - For level $h = 3$: $X = (3, 0, 5)$, $X = (0, 3, 5)$ and $X = (1, 2, 5)$
  - For level $h = 4$: $X = (1, 3, 5)$ and $X = (3, 2, 5)$
  - For level $h = 5$: $X = (3, 2, 5)$
- The minimum cut vectors:
  - For level $h = 3$: $X = (3, 3, 2)$
  - For level $h = 4$: $X = (3, 3, 2)$ and $X = (1, 2, 5)$
  - For level $h = 5$: $X = (3, 3, 2)$ and $X = (1, 3, 5)$

For a binary system, the calculation of availability by minimum paths is based on the Poincaré's formula:

$$A = \sum_{i=1}^{m} P(C_i) - \sum_{1 \le i < j \le m} P(C_i \cap C_j) + \ldots + (-1)^{m+1} P(C_1 \cap \ldots \cap C_m) \tag{3.28}$$

Such that m is the number of all the minimal paths and C_i is ith minimal path.

The extension of this formula to apply it on MSSs is:

$$A_h = \sum_{i=1}^{m} P(X \ge Y_i) - \sum_{1 \le i < j \le m} P(X \ge max(Y_i, Y_j)) + \ldots + (-1)^{m+1} P(X \ge max(Y_1, \ldots, Y_m)) \tag{3.29}$$

With m is the maximum number of the minimum paths ensuring for the level h and Y_i the minimal path vector i among the m vectors.

$$max(Y_1, \ldots, Y_m) = max(max(y_{11}, \ldots, y_{1m}), \ldots, max(y_{N1}, \ldots, y_{Nm}))$$

with $Y_1 = (y_{11}, \ldots, y_{N1})$ and N is the number of the components of the system. For example, if we have $Y_1 = (1, 2, 1)$ and $Y_2 = (3, 0, 1)$ then $max(Y_1, Y_2) = (3, 2, 1)$.

Consider a system of N components such as each component C_i can work in its levels that can take values from 0 to M_i. The maximum capacity of the system is defined by $M = \{M_1, M_2, \ldots, M_n\}$.

To define the minimum paths of the system, we will need the vector L_i which defines the levels corresponding to the states of the component C_i and the vector O_i which defines the order of these states. For example, if $L_i = (0, 3, 5, 6)$ then $O_i = (0, 1, 2, 3)$.

First, we define the primary minimum paths of the system without taking into account the states of the components. Which means, we find the minimum connections that allow to go from the source to the end point. The primary minimum path vector PMP is defined as follows, if the ith component exists in the path then the ith value of the vector takes the value 1 otherwise it takes 0.

From these PMP vectors, we will build the secondary path vectors SP from the values of L_i. It is possible to obtain vectors that are not minimal, repeatable or greater than the vector M. In this case, we choose the vectors in order to eliminate false vectors and to keep only the minimum path vectors. And finally, through the formula of Poincaré, we will be able to calculate the availability of the system for each level of performance. Here is the steps that we must do:

- **Step 1:** Find the PMP vectors, with "1" means the existence of the component in the path and "0" otherwise.
- **Step 2:** Build the primary minimum path vectors of level $h + 1$ noted PMP_{h+1} from the primary minimum path vectors of the level h noted PMP and the primary minimum paths vectors of level 1 noted PMP_1.

$$PMP_{h+1} = X + Y \text{ such that } X \in PMP_h \text{ and } Y \in PMP_1.$$

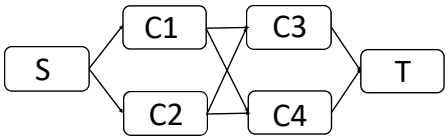


Figure 3.8: An example system

Component 1		Component 2		Component 3		Component 4	
G	**P**	**G**	**P**	**G**	**P**	**G**	**P**
0	0.1	0	0.1	0	0.1	0	0.1
2	0.1	1	0.2	1	0.1	2	0.2
3	0.8	4	0.7	3	0.8	4	0.7

Table 3.4: The characteristics of the components of the system in Figure 3.8

- **Step 3:** Merge the vectors $Z \in PMP_{h+1}$ that repeat themselves, eliminate the vectors that are not not minimal and remove vectors greater than the maximum capacity of the system M.
- **Step 4:** Repeat steps 2 and 3 for all the levels h.
- **Step 5:** Determine the minimum secondary path vectors SP from the PMPs for each level h. If the ith value of the vector PMP_h noted by x_i does not exist in the vector L_i of the component C_i, then in the vector SP_h, this value is replaced by the smallest value of the vector L_i that is higher than x_i.
- **Step 6:** Eliminate the SP vectors that are not minimal and that are repeated for each level h.
- **Step 7:** Calculate the system availability for each level h using the formula of Poincaré.

Numerical example

This example is token from the Master's report of Kaoutar Rhazali [95]. Consider the system of the Figure 3.8 and the Table 3.4 presents the characteristics of the components.

1. **State 1:**
   The vectors PMP_1 are: $X_2 = (1,0,1,0)$; $X_2 = (1,0,0,1)$; $X_3 = (0,1,1,0)$; $X_4 = (0,1,0,1)$.

The corresponding SP_1 vectors are: $\begin{cases} PMP_1 \Rightarrow SP_1 \\ (1,0,1,0) \Rightarrow (2,0,1,0) \\ (1,0,0,1) \Rightarrow (2,0,0,2) \\ (0,1,1,0) \Rightarrow (0,1,1,0) \\ (0,1,0,1) \Rightarrow (0,1,0,2) \end{cases}$

The minimum path vectors are: $X_1 = (2,0,1,0)$; $X_2 = (2,0,0,2)$; $X_3 = (0,1,1,0)$ and $X_4 = (0,1,0,2)$.

$A_1 = P(X \geq X_1) + P(X \geq X_2) + P(X \geq X_3) + P(X \geq X_4) - P(X \geq max(X_1, X_2)) - P(X \geq max(X_1, X_3)) - P(X \geq max(X_1, X_4)) - P(X \geq max(X_2, X_3)) - P(X \geq max(X_2, X_4)) - P(X \geq max(X_3, X_4)) + P(X \geq max(X_1, X_2, X_3)) + P(X \geq max(X_1, X_3, X_4)) + P(X \geq max(X_1, X_2, X_4)) + P(X \geq max(X_2, X_3, X_4)) - P(X \geq max(X_1, X_2, X_3, X_4))$

$\Rightarrow A_1 = 0.9 \times 1 \times 0.9 \times 1 + 0.9 \times 1 \times 1 \times 0.9 + 1 \times 0.9 \times 0.9 \times 1 + 1 \times 0.9 \times 1 \times 0.9 - 0.9 \times 1 \times 0.9 \times 0.9 - 0.9 \times 0.9 \times 0.9 \times 1 - 0.9 \times 0.9 \times 0.9 \times 0.9 - 0.9 \times 0.9 \times 0.9 \times 0.9 - 0.9 \times 0.9 \times 1 \times 0.9 - 1 \times 0.9 \times 0.9 \times 0.9 + 0.9 \times 0.9 \times 0.9 \times 0.9 + 0.9 \times 0.9 \times 0.9 \times 0.9 + 0.9 \times 0.9 \times 0.9 \times 0.9 + 0.9 \times 0.9 \times 0.9 \times 0.9 - 0.9 \times 0.9 \times 0.9 \times 0.9.$

$\Rightarrow A_1 = 0.9801.$

2. **Level 2:**

The vectors PMP_2 and SP_2: $PMP_2 \Rightarrow SP_2$

$\begin{cases} (1,1,1,0) + (1,0,1,0) = (2,0,2,0) \Rightarrow (2,0,3,0) \\ (1,0,1,0) + (1,0,0,1) = (2,0,1,1) \Rightarrow (2,0,1,2) \text{ (The third component is greater than the third component of the M, then this vector must be eliminated)} (2,0,0,2) \\ (1,0,1,0) + (0,1,1,0) = (1,1,2,0) \Rightarrow (2,1,3,0) \text{ (it is not a minimum vector) } (2,0,3,0) \\ (1,0,1,0) + (0,1,0,1) = (1,1,1,1) \Rightarrow (2,1,1,2) \text{ (it is not a minimum vector) } (2,1,0,2) \\ (1,0,0,1) + (1,0,0,1) = (2,0,0,2) \Rightarrow (2,0,0,2) \\ (1,0,0,1) + (0,1,1,0) = (1,1,1,1) \text{ (repeated vector)} \\ (1,0,0,1) + (0,1,0,1) = (1,1,0,2) \Rightarrow (2,1,0,2) \text{ (it is not a minimum vector) } (2,0,0,2) \\ (0,1,1,0) + (0,1,1,0) = (0,2,2,0) \Rightarrow (0,4,3,0) \\ (0,1,1,0) + (0,1,0,1) = (0,2,1,1) \Rightarrow (0,4,1,2) \text{ (it is not a minimum vector) } (0,4,0,2) \\ (0,1,0,1) + (0,1,0,1) = (0,2,0,2) \Rightarrow (0,4,0,2) \end{cases}$

The minimum path vectors are: $(2,0,3,0)$; $(2,0,0,2)$; $(0,4,3,0)$ and $(0,4,0,2)$.

$A_2 = 0.9506.$

3. **Level 3:**
   The minimum path vectors: $(3, 0, 3, 0)$; $(2, 1, 3, 0)$; $(2, 1, 3, 0)$; $(3, 0, 1, 2)$; $(3, 0, 0, 4)$; $(2, 1, 1, 2)$; $(0, 4, 3, 0)$; $(0, 4, 1, 2)$ and $(0, 4, 0, 4)$.

   $A_3 = 0.7921.$

4. **Level 4:**
   The minimum path vectors: $(3, 1, 3, 2)$; $(0, 4, 3, 2)$; $(3, 1, 0, 4)$ and $(0, 4, 0, 4)$.

   $A_4 = 0.7396.$

5. **Level 5:**
   The minimum path vectors: $(2, 4, 3, 2)$ and $(2, 4, 1, 4)$.

   $A_5 = 0.4977.$

6. **Level 6:**
   The minimum path vectors: $(2, 4, 3, 4)$.

   $A_6 = 0.3528,$

7. **Level 7:**
   The minimum path vectors: $(3, 4, 3, 4)$.

   $A_7 = 0.3136$

From these results, we can deduce the U-function of the system:

$$U_{system} = (1 - A_1) * Z^0 + (A_1 - A_2) * Z^1 + (A_2 - A_3) * Z^2 + (A_3 - A_4) * Z^3 + (A_4 - A_5) * Z^4 + (A_5 - A_6) * Z^5 + (A_6 - A_7) * Z^6 + A_7 * Z^7$$

$$U_{system} = 0.0199 * Z^0 + 0.0295 * Z^1 + 0.029 * Z^2 + 0.182 * Z^3 + 0.2419 * Z^4 + 0.1449 * Z^5 + 0.0392 * Z^6 + 0.3136 * Z^7$$

"Inclusion-Exclusion" method is to evaluate the availability of MSSs with a complex structure. It is based on the minimal paths and the Poincaré's formula. Even it is efficient, this procedure requires a lot of computing time. Also, the number of minimal path vectors

is much larger than the number of nodes and number of links in the system, on the other hand this makes the application of the Poincaré's formula much complex, especially when the number of components states increases.

3.3.4 Monte-Carlo simulations

This technique is well used in dependability [93, 94, 124, 131]. It offers a powerful means for evaluating the availability of a system, due to the flexibility in modeling that it offers regardless the type and the dimension of the problem. The idea of this method is the generation of certain random and discrete events in a computer model in order to create a realistic lifetime model of the system. Therefore the simulation of the system's life process will be carried out in the computer, and estimations will be made for the desired measures of performance [13]. The simulation will be then treated as a series of real experiments, and statistical inference will then be used to estimate confidence intervals for the performance metrics. The events can be simulated either with variable time increments (discrete event simulation) or with fix time increments, at equidistant points of time (continuous time simulation).

An approach was presented using Monte-Carlo simulation and an enumeration technique for the reliability evaluation of MSS [12]. Another work by Zio *et al.* [131], where they presented a Monte Carlo simulation technique which allows modelling the complex dynamics of multi-state components subject to operational dependencies with the system overall state. Jose et *et al.* [93] described a Monte-Carlo simulation methodology for evaluating the reliability of a MSS. Zio and Podofillini [132] have presented a Monte-Carlo simulation approach to estimate all the importance measures of the components at a given performance level in a multi-state series-parallel system.
In all of these works, we can find several examples that show how we apply Monte-Carlo simulation on MSS, however we will not present them because of the various sort of applications.

The weak point of the Monte Carlo method is the computing time. The method is based on the repeated samples of realizations of system configurations. A large number of realizations must be simulated in order to achieve an acceptable accuracy in the estimated failure probability with costly large computing times. However, there are efficient methods of convergence acceleration when we are only interested in averages.

3.3.5 Markovian approach

The Markovian approach is based on the work of Andrei Andreevich Markov (1856-1922) in probability theory. It was applied in dependability for the first time in the 1950s. Since then, many advances and extensions have been made to solve some sort of problems.

Markovian approach is very adopted in dependability to calculate the availability of an MSS [75, 85, 84, 35, 69, 70]. It is possible with this method to present all the possible states. The main benefit is its easy implementation in the form of a graph of states. In our study, we use the Markovian approach to model MSS and to calculate the availability of the system. The way of use of this method will be detailed in chapter 4.

Reminder of the Markovian model

Markov models represent a class of stochastic processes, where X is a random variable with the parameter t the time which is always positive.

$$\{X(t), t \geq 0\} \tag{3.30}$$

the time t could be discrete then we talk about Markov chain or continuous we talk about Markov process. In our work the time is continuous.

A stochastic process in which the future state of the system does not depend on the past trajectory, it is memoryless. The Markovian model evaluates the transition rates of jumping from one known state into the next logical state until, depending upon the configuration of the system being considered, the system has reached the final or totally failed state.

The states of the system can then be deduced from the states of the components by studying the interactions between components or their simple combinations. The state equations can show the evolution of the system as a function of time in terms of the probability of occupation of the state of the system $P_i(t)$, using the transition probabilities $P_{ij}(t)$ with $dP_{ij}(t) = q_{ij}(t)dt$ during an interval time dt between two states e_i and e_j.

The parameter $q_{ij}(t)$ is called the **transition rate** and it is only defined when $P_{ij}(t)$ is differentiable. The state equations of a system S are defined in a discrete state space $E = \{e_1, \ldots, e_n\}$ with e_i is the i^{th} state of the system S, taking into account the transition rates q_{ij} existing between consecutive states of the system.

Since in our study the time is continuous, we do not talk about probabilities (like the case in discrete time), however we talk about transition rates q_{ij} (could be failure rates, repair rates, or the combinations of these two) and about the transition matrix Q, where:

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{pmatrix} \tag{3.31}$$

Diagonal elements q_{ii} are defined such that:

$$q_{ii} = -\sum_{i=1, i\neq j}^{n} q_{ij} \tag{3.32}$$

thus the sum of each row of Q is equal to 0.

The equations of the states of a system S are defined in a discrete state space E by taking into account the transition rates $q_{ij}(t)$ existing between consecutive system states. By defining the probabilities that the system stays in its current state or moves to any of the possible states in E for an elementary time interval $[t, t+dt]$, we obtain a system of differential equations, called the Chapman-Kolmogorov equations.

Thus, for each state e_i

$$P_i(t+dt) = \text{P(} S \text{ in state } e_i \text{ at } t \text{ and in } [t, t+dt]) \tag{3.33}$$

$$+ \sum_{e_j \in E - e_i} \text{P(} S \text{ in state } e_j \text{ at } t \text{ and in } e_i \text{ at } [t, t+dt]) \tag{3.34}$$

Which means:

$$P_i(t+dt) = P_i(t) q_{ii}(t) dt + \sum_{e_j \in E - e_i} P_j(t) q_{ji}(t) dt \tag{3.35}$$

We can now write down:

$$P_i(t+dt) = P_i(t)(1 - \sum_{e_j \in E - e_i} q_{ij}(t) dt) + \sum_{e_j \in E - e_i} P_j(t) q_{ji}(t) dt \tag{3.36}$$

We have:

$$\frac{P_i(t+dt) - P_i(t)}{dt} = -P_i(t) \sum_{e_j \in E - e_i} q_{ij}(t) + \sum_{e_j \in E - e_i} P_j(t) q_{ji}(t) \tag{3.37}$$

And finaly we obtain:

$$\dot{P}(t) = P(t).Q \tag{3.38}$$

with vector P represents the probabilities of being in a state in a certain time t

$$\begin{cases} P = \begin{bmatrix} P_1(t) & \dots & P_n(t) \end{bmatrix} \\ \sum_{j=1}^{n} P_j(t) = 1. \end{cases} \tag{3.39}$$

In our study, we have Q is independent from t, since we take the case of constant transition rates (failure rates and repair rates).

In many applications, only the steady-state probabilities are interesting, since we want to know if the product is survivable till when. The steady-state probabilities is the values of $P_j(t)$ when $t \longrightarrow \infty$.

The process is said to be irreducible if every state is reachable from every other state. For an irreducible Markov process, it can be shown that the limits

$$\lim_{t \longrightarrow \infty} P_j(t) = \pi_j \text{ for } j = 1, 2, \dots, n$$

always exist and are independent of the initial state of the process (at time $t = 0$).

The asymptotic probabilities are often called the steady-state probabilities for the Markov process.

If $P_j(t)$ tends to a constant value when $t \longrightarrow \infty$, then:

$$\lim_{t \longrightarrow \infty} \dot{P}_j(t) = 0 \text{ for } j = 1, 2, .., n$$

The steady-state probabilities $\Pi = [\pi_1, \pi_2, \dots, \pi_n]$ must therefore satisfy the matrix equation:

$$[\pi_1, \pi_2, \dots, \pi_n]. \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ .. & .. & \dots & .. \\ q_{n1} & .. & \dots & q_{nn} \end{bmatrix} = [0, 0, \dots, 0] \tag{3.40}$$

which may be abbreviated by:

$$\Pi.Q = 0 \tag{3.41}$$

and this equation will be the main equation that we will use later in the aim to solve our problem. With as before:

$$\sum_{j=1}^{n} \pi_j = 1$$

The main idea is to find the solution Π so that we could calculate the availability of the system when t tends to infinity.
In this aim, we define the input:

$$\mathbf{P}(0) = \begin{bmatrix} P_{10} & P_{20} & \dots & P_{n1} \end{bmatrix} \tag{3.42}$$

which is the initial vector of the states of system. From the data of all the components (failure rates and repair rates), we form our transition matrix Q, and we solve the system of equation in eq 4.1 to get Π.

3.4 Imprecise availability computing

In the real world of MSSs, there is an insufficiency of data which makes it difficult to estimate precise values of component's failure rates, repair rates and state probabilities [100]. Nowadays, usually the estimation of the effect of uncertainty is a necessity, e.g. due to variation in parameters, operational conditions and in the modeling and simulations [9].

Uncertainties are classified into two categories: epistemic uncertainty and aleatory uncertainty [100]. Epistemic uncertainty is due to the lack of knowledge of quantities. Aleatory uncertainty is due to the inherent variation associated to the physical system [100].

Uncertainties was introduced in all kind of works in dependability field. In [2], they use the belief function theory on the Inclusion-Exclusion method in dependability. In [129], they used intervals to model imprecise probability using Monte-Carlo method. In this section, we talk about some of the methods that we will use them later in chapter 4. In these methods, they applied uncertainty in the calculation of imprecise availability.

3.4.1 IUGF and BUGF

BUGF (Belief Universal Generated Function) [28] and IUGF (Interval Universal Generated Function) [66] are two methods based on the UGF (Universal Generated Function) but they are applied on intervals and therefore are used in the case of interval-modeled imprecision. These two methods are efficient and give good results, the BUGF is also noted as more efficient than the IUGF.

IUGF

In [66], a method is proposed to analyze the reliability of MSSs when the available data of components are insufficient. Based on the Bayesian approach and the imprecise Dirichlet

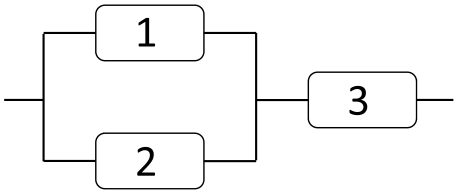


Figure 3.9: Flow transmission system

model, the state probabilities of components are obtained as intervals instead of precise values. They developed the interval universal generating function, and the corresponding operators are defined to estimate the reliability of MSSs in interval form. They used affine arithmetic to find the interval-valued reliability. The results show the efficiency of the method when the performance levels of states and/or the state's probabilities of components are uncertain.

BUGF

In [28], they proposed a method which is an extension of the UGF method taking into account epistemic uncertainties. This method allows to model ill-known probabilities and transition rates. Based on the use of belief functions which are general models of uncertainty they developed their method. In this work, they also compared this extension with UGF methods based on interval arithmetic operations performed on probabilistic bounds like the work cited previously [66].

To compare the two methods, they gave a numerical example that was presented in this article and in [66]. The example is from Yi Ding and Anatoly Lisnianski [29]. Where they changed the fuzzy into interval numbers in [66]. As Figure 3.9 shows, the flow transmission system consists of three components. It is supposed that components 1 and 2 have three possible states: a state of total failure corresponding to a capacity of 0, a state of full capacity, and a state of partial failure. The component 3 only has two states: a state of total failure, and a state of full capacity. All state performances of the components are precise. The state probabilities state p_i^j and the performance levels g_i^j of the components of the system are given in Table 3.5.

- State 3 of component j: g_3^j represents the performance rate of the completely successful state.
- State 2 of component j: g_2^j represents the performance rate of the degraded successful state.

Component j	1	2	3
p_1^j	[0.096, 0.106]	[0.095, 0.105]	-
p_2^j	[0.095, 0.105]	[0.195, 0.205]	[0.032,0.042]
p_3^j	[0.799,0.809]	[0.7, 0.71]	[0.958, 0.968]
g_1^j	0	0	-
g_2^j	1	1.5	0
g_3^j	1.5	2	4

Table 3.5: Parameter of the components of the system

- State 1 of component j: g_1^j represents the performance rate of the total failure state.

and p_i^j is the probability of being in state g_i^j. Here the procedures and the results by each method. *A. Interval UGF approach (IUGF)*

According to Li *et al.* [66], the IUGF of a component j with M_j states is defined by:

$$U_j(z) = \sum_{i=1}^{M_j} [p_i^j].z^{g_i^j} \tag{3.43}$$

The IUGF of a system with n components is obtained as follows:

$$\begin{aligned} U(z) &= \omega(U_1(z), \ldots, U_n(z)) \\ &= \omega(\sum_{l_1=1}^{k_1} [p_{l_1^1}].z^{g_{l_1}^1}, \ldots, \sum_{l_n=1}^{k_n} [p_{l_n^n}].z^{g_{l_n}^n}) \\ &= \sum_{l_1=1}^{k_1} \sum_{l_2=1}^{k_2} \cdots \sum_{l_n=1}^{k_n} [p_{l_1^1}].[p_{l_2^2}] \ldots [p_{l_n^n}] z^{\phi(g_{l_1}^1, g_{l_2}^2, \ldots, g_{l_n}^n)} \\ &= \sum_{l_1=1}^{k_1} \sum_{l_2=1}^{k_2} \cdots \sum_{l_n=1}^{k_n} [\prod_{i=1}^{n} \underline{p_{l_i}^i}, \prod_{i=1}^{n} \overline{p_{l_i}^i}] z^{\phi(g_{l_1}^1, g_{l_2}^2, \ldots, g_{l_n}^n)} \\ &= \sum_{i=1}^{R} [p_i].z^{r_i} \end{aligned} \tag{3.44}$$

where $\phi(g_{l_1}^1, g_{l_2}^2, \ldots, g_{l_n}^n)$ depends on the component states. The final line summarizes the last big sum into a sum of probabilities over system performance levels. For a demand level w, the system availability $[A]$ is obtained such that:

$$[A] = \sum_{i=1}^{R} [p_i | r_i \geq w] \tag{3.45}$$

Using eq. 3.43, the IUGF of the three components are:

$$\begin{cases} U_1(z) = [0.799, 0.809]z^{1.5} + [0.095, 0.105]z^1 + [0.096, 0.106]z^0 \\ U_2(z) = [0.7, 0.71]z^2 + [0.195, 0.205]z^{1.5} + [0.095, 0.105]z^0 \\ U_3(z) = [0.958, 0.968]z^4 + [0.032, 0.042]z^0 \end{cases}$$

The IUGF of the subsystem formed by components 1 and 2 is

$$U_{12}(z) \begin{cases} = \omega(U_1(z), U_2(z)) \\ = [0.55593, 0.5744]z^{3.5} + [0.2223, 0.24035]z^3 + [0.0185, 0.00215]z^{2.5} + [0.0672, 0.075]z^2 \\ +[0.0946, 0.1066]z^{1.5} + [0.009, 0.011]z^1 + [0.0091, 0.0111]z^0 \end{cases}$$

We obtain the overall IUGF of the system by applying:

$$U_{123}(z) = \omega(U_{12}(z), U_3(z))$$

To estimate the availability of the system when $w = 1.5$, we use eq. 3.45 and we obtain:

$$[A] = [0.9183, 0.9665]$$

B. Belief UGF approach (BUGF)

They applied the UGF extension to the system of 3 components. To apply Belief function, mass functions of probability intervals described in Table 3.5 can be obtained using the following formula

$$m(E) = \begin{cases} \underline{p_k^j} \text{ if } E = g_k^j \\ 1 - \sum_{k=1}^{k_j} \underline{p_k^j} \text{ if } E = G^j \\ 0 \text{ else} \end{cases} \tag{3.46}$$

with E presents the set of the performance level and G^j is the set of all the performance levels of the component j.

The BUGF equations of the three components is obtained by using

$$U_j(z) = \sum_{i=1}^{F_j} m(g_i^j).z^{g_j^i} \tag{3.47}$$

with F_j is the number of the performance level of the component j.
We have:

$$\begin{cases} U_1(z) = 0.799z^{\{1.5\}} + 0.095z^{\{1\}} + 0.096z^{\{0\}} + 0.01z^{\{0,1,1.5\}} \\ U_2(z) = 0.7z^{\{2\}} + 0.195z^{\{1.5\}} + 0.095z^{\{0\}} + 0.01z^{\{0,1.5,2\}} \\ U_3(z) = 0.958z^{\{4\}} + 0.032z^{\{0\}} + 0.01z^{\{0,4\}} \end{cases}$$

The overall BUGF of the system is computed similar as in IUGF, we found out $U_{12}(z)$ using $U_1(z)$ and $U_2(z)$.Then we use the min operator to $U_{12}(z)$ with the UGF of component 3.

Concerning the availability of the event $A = r_i, \ldots, r_R$ with $w = 1.5$, they defined two operators δ_A^+ and δ_A^- that compute plausibility and belief values of A.

$$\begin{aligned} \delta_A^+(U(z)) &= \delta_A^+(\sum_{i_1=1}^{F_1}\sum_{i_2=1}^{F_2}\cdots\sum_{i_n=1}^{F_n}\prod_{j=1}^{n} m_{i_j}^j z^{\phi([g]_{i_1}^1,\ldots,[g]_{i_n}^n)}) \\ &= \sum_{i_1=1}^{F_1}\sum_{i_2=1}^{F_2}\cdots\sum_{i_n=1}^{F_n}\prod_{j=1}^{n} m_{i_j}^j \delta_A^+(z^{\phi([g]_{i_1}^1,\ldots,[g]_{i_n}^n)}) \end{aligned} \tag{3.48}$$

with

$$\delta_A^+(z^{\phi([g]_{i_1}^1,\ldots,[g]_{i_n}^n)}) = \begin{cases} 1 \text{ if } r_i \leq_{\mathbf{R}} \phi([g]^1,\ldots,[g]^n)^+ \\ 0 \text{ else} \end{cases} \tag{3.49}$$

we obtain δ_A^- by replacing the plus signs by minus signs. $\mathbf{R}$ is the set of global performance rates of the system and $\leq_{\mathbf{R}}$ is the order relation on $\mathbf{R}$.

The availability interval can be defined, it contains the precise availability, and is bounded by two non-additive continuous measures called belief (or support) and plausibility. Belief function $Bel(A)$ is the amount of belief that directly supports either the given hypothesis or a more specific one, thus forming a lower bound on its probability. It is the sum of all the masses that support A. Plausibility function $Pl(A)$ is an upper bound on the possibility that the hypothesis could be true and it represents the total amount of masses that might support A.

We have that $\delta_A^+ = Pl(A)$ the plausibility and $\delta_A^- = Bel(A)$ the belief function.
This can be seen by noticing that $r_i \leq_{\mathbf{R}} \phi([g]^1,\ldots,[g]^n)^+$ means that at least one element of the interval-valued performance $\phi([g]^1,\ldots,[g]^n)$ is above (or equal to) performance r_i, hence $\phi([g]^1,\ldots,[g]^n) \cap A \neq 0$, while if $r_i \geq_{\mathbf{R}} \phi([g]^1,\ldots,[g]^n)^+$, no elements of $\phi([g]^1,\ldots,[g]^n)$ is in A. Similarly, $r_i \leq_{\mathbf{R}} \phi([g]^1,\ldots,[g]^n)^-$ is equivalent to $\phi([g]^1,\ldots,[g]^n) \subseteq A$ [28].

We get:

$$[Bel(A), Pl(A)] = [0.9377, 0.9505]$$

3.4.2 Imprecise Markovian model

When the initial states probabilities and the transition rates of a finite Markov chain in discrete time are not well known, we should talk about imprecise Markov model.

The Markov assumption stating that X_{t+dt} is conditionally independent of X_s, for $s < t$, knowing X_t may not be realistic, especially for repair. Also, the transition rates may not be constant in time, but are usually affected by a variety of factors, and the estimation of the rates themselves may be difficult due to the lack of data.

A full modeling of these details requires a lot of data and expert knowledge. Instead of ignoring this problem, a better way to cope with it is to incorporate the imprecision into the models. This becomes possible with the development of models of imprecise probabilities.

Our contribution will be based on imprecise Markov model, in chapter 4 we will give a detailed explanation of this concept with the related work.

3.5 Problem statement and positioning

A MSS, as we said before, is a system of components and each of these components could have more than one state of functioning. Our main point, is to calculate the availability of such system. In this thesis, we assume that the components are independent components and that the only influence of the environment is that of repair/control and demand (system solicitation). In particularly, the change of the component's state depends only the components (no influence of other components or the environment), except maintainability.

Finding the availability of an MSS with complex structure isn't simple. A system with complex structure presents all the possible combinations between the elements, without respecting one type of connection. It becomes much complicated when the system is larger (number of components $n > 60$).

The total number of component's states increases when the number of components increases and when the system become more complex. Therefore, the combination between all the possible states become much complicated. The Inclusion-Exclusion method treats the case where all components are binaries, it studies a particular case, thus we should not use it in our work. Also, the UGF method is applied only if the system is simple (parallel, series, series-parallel,...), which means it could not be used when a system is complex and

this isn't our case. Thus, this proposed method costs a lot in terms of time.

We should also take into account that in dependability studies, the information about the data of the components are generally imprecise. The obtained results depend on the imprecision of the data used. The uncertainties result from the model itself which means it isn't perfect. Also, the performance values obtained could be imprecise or wrong. The data that we have (failure rates λ and repair rates μ) aren't exact because the estimation of the transition rates is difficult (new elements, rarely affected elements, expensive elements,...) or they may not be constant in time. It requires some methods to model and manipulate the imprecision like the probability theory, the fuzzy theory, the belief function theory,... Some of the works presented in this chapter, have been done to calculate the imprecise availability of MSS, we mention the work of Li.[66] where he used the arithmetic interval calculation applied to UGF (IUGF, interval universal generating function) and the work of Destercke et Sallak [28] where they proposed to apply belief functions on the UGF (BUGF, belief universal generating function). These two methods study the case of system with imprecision, since they are based on the UGF method, they don't treat the case of complex system, which means we need another method to take into account the complexity of the system and the imprecision. For these two methods we need first to transform the failure rates and the repair rates in terms of probabilities then apply the technique. Their use is limited to the case of systems with simple configuration and the imprecise availability assessment will cost a lot of time.

Many methods exist to find the availability of the system, however in this thesis, we choose to work with Markovian models applied on the complex MSS. With Markovian models, it is easier to model the system with its sequence of events such as breakdowns, repairs,... It is very flexible in the type of systems and system behavior it can model [16]. In general, two states are considered for a component of the system (working/ failing) it is possible to present more (degraded, repair,...). The main benefit, is its easy implementation in the form of a graph of states. The number of states that grows with the number of components and therefore makes it difficult to assess the availability but approximations exist for large sizes of systems, especially when considering asymptotic availability, which is our aim.

However, as we said before, we are dealing with uncertainties on the given data (failure rates, repair rates, the states probabilities). Therefore, we decide to model these data in interval form and introduce Interval analysis to this work. Intervals represent each value

as a range of possibilities. We keep track of and handle rounding errors directly during the calculation and of uncertainties in the knowledge of the exact values of physical and technical parameters. Therefore, in our case, we will have imprecise Markov model. Imprecise Markovian model is the most general method, since we are able to model an MSS with all the possible combination even if the system is complex with the presence of imprecision. We should take into account later, that we may have some difficult when the number of components increases a lot. In this case, we will apply some approximations. To our best of knowledge, there is one work dealt with this case which will be presented in chapter 4.

To calculate the availability of the system, we will apply the imprecise Markov model, which is defined in the previous section so we could solve the eq. 4.1 and find the vector Π. Solving this system of equation in the aim to compute Π, isn't simple. Since we are dealing with intervals (matrix, vector,...), we can use a technique called "the technique of contractors". This method is well used in interval analysis, and it will be applied in dependability for the first time. The main idea is to contract each interval as much as we can using some operators called "contractors", which will give us a guaranteed result.

3.6 Conclusion

In this chapter, we presented some of the related work for availability assessment. We started with the case of binary systems then we pass to the case of MSSs. After that we introduced uncertainties and we presented some of the related works with this condition.

Part 2 deals with our contributions. In chapter 4, we will present our first contribution where we will present our methodology in the aim of calculating the imprecise availability of an MSS using Markovian approach. In chapter 5, we will present how we can optimize the imprecise availability of an MSS taking into account the cost factor of the system.

Part II

Contributions

Chapter 4

Imprecise availability estimation of MSS

4.1 Introduction

A system as defined in the introduction of chapter 1, is any group of at least two interacting or interrelated elements that form a unified whole such that there are no independent elements [119]. Each system is designed to perform a specific function. This function is defined by the connections that these components make between each other. As long as the connections between the components are simple and can be defined, the system has a simple structure. As the connections become more complex, the system has a complex structure. In addition to the complex structure of the system, the system and the components can have multi-states.

Computing the availability of a multi-states system (MSS) with complex structure isn't simple. Although, the MSSs with complex structures represent the most general case of systems that may exist and the binary case is a particular case from the complex cases. Still the computing of their availability remains difficult and complicated. This calculation becomes more complicated when the system is larger, which means when the number of components increases. Since, the complex system cannot be decomposed into subsystems with simple structures, we must find and use different strategies that help us in this case. Also, even if methods that estimate uncertainties on data (failure rates, repair rates, initial probabilities to be in a given state) do exist, but this estimation is sometimes hard to get (components rarely affected, expensive components, new components, . . .). An objective is to develop efficient method that can deal with the uncertainties problem.

When studying the availability of a system, one may face some difficulties due to the fact that the system can be an MSS, which means that the system and its components may

have different states. In our work, we assume imprecise failure rates and repair rates; this requires methods that allow to model and to manipulate these uncertainties. In this thesis, we are interested in uncertainties about reliability data (failure rate λ and repair rate μ of each component). The proposed method to cope with these uncertainties is based on the theory of imprecise probabilities, in particular, interval probabilities and the corresponding interval analysis (the choice of this approach is discussed in section 3.5).

Interval analysis is a branch of the numerical analysis which uses for calculation closed intervals of real numbers instead of precise numbers [30]. Interval analysis is used in order to obtain computational results that are surely true. The result of an ordinary computation is a single number, a point on the real line, which lies at some distance from the true answer. In this context, a question may arise: how large is this distance? To answer this question, we must evaluate the data in a marge of values which is the interval. The result of an interval computation is an interval, a pair of numbers, an upper and a lower bounds, and this pair of numbers guarantees to enclose the exact value. In our case, we consider that uncertain transition rates are bounded by intervals, $\lambda = [\underline{\lambda}, \overline{\lambda}]$ and $\mu = [\underline{\mu}, \overline{\mu}]$.

To model the complex structure of the MSS with its complex structure, we propose the use of Markov model and calculate its asymptotic availability by first solving the following system of equations:

$$\Pi.Q = 0 \tag{4.1}$$

with $\Pi = [[\pi_1] \ \ldots \ [\pi_n]]$ is the state's probability vector of the system, where its elements are $[\pi_i]$ the probability interval of the system to be in the *i*th state. Q is the interval transition matrix. The availability is then equal to the sum of the probabilities related to the functioning state.

Markovian models are generally used in dependability domain. They are commonly used for modeling large systems and for the evaluation of the performance and dependability of complex systems. They present, in a logical and simple way, the transitions from one operating state to another one. Two major problems that are encountered in the use of Markov models are largeness and stiffness [84]. Complex systems induce large and complex Markov models. The largeness problem can be addressed by either avoiding it through aggregation and decomposition, or by using automated methods for generating the large and complex Markov chains. Stiffness often results from having transition rates of different orders of magnitude in the Markov chain. The problem of stiffness is avoided by using special methods [84].

In this thesis, we will propose a new method that can calculate the imprecise availability in a simple way. The availability of an MSS that we are interested in, is the probability that the system remains working after a period of time. In a simple way, we are looking for the imprecise steady probabilities of the system's states so that we can evaluate the imprecise availability of the system. Markovian models are usually used to model the transition states in a transition matrix Q and to calculate the steady probabilities of each state. In this thesis, we propose to handle imprecise data as intervals and to apply interval analysis on imprecise Markovian approaches, thus we could calculate the imprecise availability of a system. To calculate the interval form of the availability, we will recall "the technique of contractors" defined in chapter 2. In particular, we will use the contractor of the forward-backward propagation, in order to contract the intervals of steady probabilities and to assess efficiently the imprecise availability of the studied system. Note that, in our study we may have the problem of largeness. In this case, we can apply some approximations in order to simplify the problem.

In this chapter, first we detail the imprecise Markovian model in dependability. Then, we present the original work handling the use of imprecise Markovian model in dependability where we show the bad side of it. Later, we introduce the method that we call "Exact method" and we propose our contribution in the aim of estimate the imprecise availability of an MSS. This method takes all the constraints into account in the aim of computing the availability of a complex MSS.
To understand all the steps of the methodology, numerical examples will be presented later and compared to other results obtained by different existing methods (IUGF (cf. section 3.4), BUGF (cf. section 3.4), Exact method). We end this chapter with discussions and conclusions.

4.2 The use of imprecise Markovian models in dependability studies

In chapter 3, we talked about the Markovian model and we presented its basics and some of the related works that have been done. In this section, we focus on the imprecise Markovian model and how it is used in our contribution.

In dependability studies, in order to assess availability using Markov model, it is neces-

sary to calculate the probabilities of functioning for each state of the system. These probabilities are usually considered as precise and perfectly known. It is also supposed that all information on the behavior of the system and its components concerning reliability are known. As Utkin wrote in [114], it assumes two main conditions:

- All probabilities or probability distributions are known or perfectly determinable.
- The system components are independent, i.e. all random variables which describe the component reliability behavior are independent or alternatively, their dependence is precisely known.

In real systems, the first condition is rarely fulfilled [114]. In general, the reliability assessments that are integrated to describe systems and components come from different and various sources [103]. Some information may be objective measures of relative frequencies or established models. Other information may be provided by experts who usually incorporate epistemic uncertainty. The problem of uncertainty in dependability domain, in our case for availability computing, has led to several approaches. In this work, we have an MSS with multi-states components where the systems have a complex structure and uncertainties on the data. To solve the problem of finding the availability, we use imprecise Markovian model where we have imprecise probabilities. These imprecise probabilities are be treated in terms of intervals (cf. chapter 3, section 3.5).

Before passing to the next section, leading with is the methodology of the imprecise availability computing for a MSS, we need to recall the basics definitions and procedure of the imprecise Markovian model. In the next subsection, we give some of the related work on imprecise Markovian model. Later, we explain a technique of computing the imprecise availability, that we call "Exact technique". Finally, we propose our methodology and we will explain it.

4.2.1 Basics of imprecise Markovian model

The Markov assumption stating that X_{t+dt} is conditionally independent of X_s, for $s < t$, knowing X_t may not be realistic, especially for repair, also the transition rates may not be constant in time, but are usually affected by a variety of factors, and the estimation of the rates themselves may be difficult due to the lack of data. Particularly, under constant transition rates, repair times are exponentially distributed and are independent of the history of the system, but repairs will often follow a binomial distribution rather than an exponential distribution; the same applies for failure rate.

A full modeling of these details requires a lot of data and expert knowledge. Instead of ignoring this problem, a better way to cope with it is to incorporate the imprecision into the models. This becomes possible with the development of models of imprecise probabilities, such as the interval probability model. It seems therefore convenient to consider our transition rates as not being precise, but instead being bounded by an interval. Imprecision may also exist on the initial probabilities or the transition matrix, and sometimes even on both. To model this imprecision, probabilities will be replaced by intervals [1]. Thus, we have the following imprecise Markov model:

$$\begin{cases} P(X_0 = i_0) = [\underline{p_i}, \overline{p_i}] \\ P(X_n = j = i_n | X_{n-1} = i = i_{n-1}, \ldots, X_0 = i_0) = P(X_n = j | X_{n-1} = i) \end{cases} \tag{4.2}$$

With $x = \{1, \ldots, N\}$, i and j are two elements of x. X_0, X_{n-1} and X_n are random variables that belong to x and satisfies eq. 4.2.

According to eq. 4.2, any probability that satisfies $p_i \in [\underline{p_i}, \overline{p_i}]$ for each state with $i \in \{1, \ldots, n\}$, can be considered as an initial distribution, and similarly for the transition matrix $Q \in [\underline{Q}, \overline{Q}]$ can be the transition matrix at time t.

We are interested to find the probability of a system to be in a working state at the infinity, thus our purpose is to compute the asymptotic availability. Under the assumptions in eq. 4.2, the stationarity of the system is hence determined in form of a vector of intervals $[\Pi] = [[\pi_0] \ldots [\pi_i] \ldots [\pi_n]]$ with $i \in \{1, \ldots, n\}$, where $[\pi_i]$ is the probability interval of being in a state e_i. Finding the interval of each stationarity state is not as simple as it might seem. In the case of precise data, we have a precise transition matrix and we can find the answer by solving a system of equations, but in the case of imprecise data, the bounds of the intervals of stationarity states cannot be obtained just by taking into account the two bounds of the transition matrix as the work presented in [110] suggests. This will be discussed in details in the next subsection.

Several methods have been proposed to solve the previous problem by finding the solution of equation 4.1 in case of imprecise data. The exact method is a technique that consists on finding all the possible transition matrices of a system and where we have to solve 4.1 and to find a vector of stationary probabilities. In order to form at the end a vector of intervals which contains all of the possible vectors. This method gives an accurate result but

its complexity increases drastically with the system's size, since it computes all the possibilities for the different transition matrices. BUGF (Belief Universal Generated Function) [28] and IUGF (Interval Universal Generated Function) [66] are two methods based on the UGF (Universal Generated Function), and are applied on intervals. Therefore, they are used in the case of interval-modeled imprecision. These two methods are efficient and give good results. The BUGF is also noted as more efficient than the IUGF, but their use is limited to the cases where the system has not a complex structure (series-parallel and parallel-series configurations). In our approach we propose to determine the asymptotic availability of the MSS by using a new technique applied on intervals, that is the technique of contractors.

4.2.2 Related work to imprecise Markovian model

To our best of knowledge, there is only one work that have been done in dependability, for availability computing of an MSS based on the imprecise Markov models, we will focus on this work and show that a part of it is wrong. This paper have been published by Troffaes *et al.* [110], where they used imprecise continuous time Markovian models for assessing the reliability of power networks. They explore how imprecise continuous time Markovian models can improve traditional availability models based on precise continuous time Markov models. They analyzed the availability of power networks under very weak statistical assumptions, explicitly accounting for non-stationary failure and repair rates and the limited accuracy by which common cause failure rates can be estimated. Bounds on typical quantities of interest are derived, namely the expected time spent in system failure state, as well as the expected number of transitions to that state.

Modeling repair requires much more sophisticated mathematical methods which have been only very recently developed for imprecise continuous time Markov chains [104]. Since failure rates often follow a so-called bathtub curve due to burn-in and wear-out effects, and can be affected in quite complex ways by the repair history of the system, a full modeling of these details requires a lot of data and expert knowledge. Therefore, they consider that the transition rates as not being precise, but instead being bounded by an interval, to cover a range of distributions that is more likely to occur in reality, without having to be too precise about the details of this distribution, or on how this distribution depends on the history of the system.

Following [104], one should make a discretization of imprecise continuous time Markov chain and should use lower and upper transition operators [24]. The imprecise probability distributions over the set of states are calculated in terms of the corresponding expectation

operators. This problem was translated to an interval differential equation whose boundary solutions define the required values of the imprecise functional. In this framework, practical calculations such as calculating lower and upper long run probabilities can be done via linear programming [104]. An approximation is applied in order to find the expected number of times that the system visits the totally failed state, as well as the expected amount of time that it spends there, in a given time period. For the imprecise case, simple bounds on these quantities are derived.

In this work, to solve the problem of uncertainties the authors considered that failure rates and repair rates are bounded by intervals. But it isn't the case for the transition matrix $[Q]$. Instead, they found out two matrices, $\underline{Q}$ the lower bound of $[Q]$ and $\overline{Q}$ the upper bound of $[Q]$. The transition rates $[q_{ij}]$ depending on the intervals of the failure rates and the intervals of the repair rates of the components of the system. $\underline{Q}$ is obtained by finding its elements $\underline{q_{ij}}$, where $\underline{q_{ij}}$ are the lower values of $[q_{ij}]$. Contrariwise, $\overline{Q}$ is obtained by finding its elements $\overline{q_{ij}}$, where $\overline{q_{ij}}$ are the higher values of $[q_{ij}]$. By obtaining two matrices, the problem is turned into two precise cases of Markovian model. By solving $\Pi.Q = 0$, the two vectors of $[\Pi]$ are founded, the probability vector of the system to be in a specific state; the lower bound $\underline{\Pi}$ and the upper bound $\overline{\Pi}$. In this way, we can find the interval of the availability of system, formed from two bounds: the lower bound is the sum over all the elements of $\underline{\Pi}$ corresponding to working states, and the upper bound is the sum over all the elements of $\overline{\Pi}$ corresponding to working states.

To more illustrate this methodology, let us analyze the example proposed by the authors. Consider a simple network consisting of just two power lines, called A and B. The continuous time Markov chain modeling this system as follows. The state space is $E = \{AB, A, B, \emptyset\}$, where the labels of the states denote the non-faulty components (i.e. both A and B are non-faulty in AB, whereas both are faulty in $\emptyset$). They model common cause failures by assigning all failures to any one of the following three events:

- AI: independent failure of A.
- BI: independent failure of B.
- CAB: common cause failure of both A and B.

To simplify, any interval $[x]$ will be written as x.
Using standard notation from the literature on common cause failure modeling, we denote by q_1^A the rate of AI, q_1^B the rate of BI and q_2 the rate of CAB. Similarly, let r_A be the repair

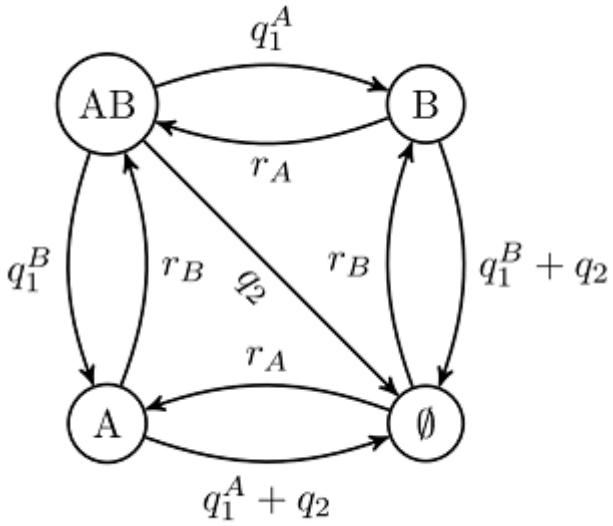


Figure 4.1: Troffaes et al. example: Markov chain for failure with non-instant repair. The nodes present the states of the system.

rate of A and r_B the repair rate of B, for simplicity they excluded simultaneous repair. The rate matrix is then:

$$[Q] = [\underline{Q}, \overline{Q}] = \begin{bmatrix} -q_1^A - q_1^B - q_2 & q_1^B & q_1^A & q_2 \\ r_B & -q_1^A - q_2 - r_B & 0 & q_1^A + q_2 \\ r_A & 0 & -q_1^B - q_2 - r_A & q_1^B + q_2 \\ 0 & r_A & r_B & -r_A - r_B \end{bmatrix} \tag{4.3}$$

The corresponding digraph of the continuous time Markov chain is depicted in Figure 4.1.

For q_1^A, q_1^B and q_2, the authors used the data and intervals for failure rates derived in the example in [111], under the approximate assumption of immediate repair, which seems reasonable as the system will spend most of its time in state AB. In these data, A and B are two identical distribution lines, and the intervals for the expected failure rates are:

$$\begin{aligned} q_1^A &= [0.32, 0.37] \\ q_1^B &= [0.32, 0, 37] \\ q_2 &= [0.19, 0, 24] \end{aligned} \tag{4.4}$$

expressed per year. In this study, there is no repair time data. Through expert elicitation, the repair rates are judged between 6 and 12 hours to be reasonable:

$$\begin{aligned} r_A &= [730, 1460] \\ r_B &= [730, 1460] \end{aligned} \tag{4.5}$$

expressed per year.

The transition matrix Q, is the interval $Q = \left[\underline{Q}, \overline{Q}\right]$. The lower bound $\underline{Q}$ is defined by:

$$\underline{Q} = \begin{bmatrix} -\overline{q_1^A} - \overline{q_1^B} - \overline{q_2} & \underline{q_1^B} & \underline{q_1^A} & \underline{q_2} \\ \underline{r_B} & -\overline{q_1^A} - \overline{q_2} - \overline{r_B} & 0 & \underline{q_1^A} + \underline{q_2} \\ \underline{r_A} & 0 & -\overline{q_1^B} - \overline{q_2} - \overline{r_A} & \underline{q_1^B} + \underline{q_2} \\ 0 & \underline{r_A} & \underline{r_B} & -\overline{r_A} - \overline{r_B} \end{bmatrix} \tag{4.6}$$

by replacing each term by its value, we obtain:

$$\underline{Q} = \begin{bmatrix} -0.98 & 0.32 & 0.32 & 0.19 \\ 730 & -1460.61 & 0 & 0.51 \\ 730 & 0 & -1460.61 & 0.51 \\ 0 & 730 & 730 & -2920 \end{bmatrix} \tag{4.7}$$

and its upper bound $\overline{Q}$ is defined by:

$$\overline{Q} = \begin{bmatrix} -\underline{q_1^A} - \underline{q_1^B} - \underline{q_2} & \overline{q_1^B} & \overline{q_1^A} & \overline{q_2} \\ \overline{r_B} & -\underline{q_1^A} - \underline{q_2} - \underline{r_B} & 0 & \overline{q_1^A} + \overline{q_2} \\ \overline{r_A} & 0 & -\underline{q_1^B} - \underline{q_2} - \underline{r_A} & \overline{q_1^B} + \overline{q_2} \\ 0 & \overline{r_A} & \overline{r_B} & -\underline{r_A} - \underline{r_B} \end{bmatrix} \tag{4.8}$$

and we will have:

$$\overline{Q} = \begin{bmatrix} -0.83 & 0.37 & 0.37 & 0.24 \\ 1460 & -730.51 & 0 & 0.61 \\ 1460 & 0 & -730.51 & 0.61 \\ 0 & 1460 & 1460 & -1460 \end{bmatrix} \tag{4.9}$$

They evaluated the lower and upper stationary distributions by using the above two transition matrices, and solving $\underline{\Pi}.\underline{Q} = 0$ and $\overline{\Pi}.\overline{Q} = 0$, without taking into account that the sum of the steady probabilities is equal to one ($\sum \pi_i = 1$). For the stationary distribution, the authors found:

$$\underline{\Pi}_1 = \begin{bmatrix} 9.985 \times 10^{-1} \\ 2.623 \times 10^{-4} \\ 2.623 \times 10^{-4} \\ 6.513 \times 10^{-5} \end{bmatrix} \tag{4.10}$$

$$\overline{\Pi}_1 = \begin{bmatrix} 9.994 \times 10^{-1} \\ 7.252 \times 10^{-4} \\ 7.252 \times 10^{-4} \\ 1.647 \times 10^{-4} \end{bmatrix} \tag{4.11}$$

such that $[\underline{\Pi}_1, \overline{\Pi}_1]$ is the interval of Π obtained by the authors method. To find the interval vector $[\Pi]$, it is by regrouping the two bounds ($\underline{\Pi}$ and $\overline{\Pi}$) together. The authors could find the availability of the system from the interval vector $[\Pi]$. In this example, we find by doing the sum over π_1, π_2, π_3 that the availability is: $[A_1] = [0.9990246, 1.00085]$

In this work, the authors looked at a model for dealing with common cause failures in power networks with two power lines, where intervals for the failure and repair rates are used to allow us to make accurate yet robust prediction of behavior under relatively weak statistical assumptions. Using imprecise Markov chains allows for the case where failure and repair rates are not constant in time, and allows for them to properly capture the uncertainty regarding common cause failures which are very hard to quantify. For all these reasons, imprecise continuous time Markov chains have a lot of potential to improve traditional reliability models based on precise Markov chains. However, they made a mistake in this work. The lower transition matrix $\underline{Q}$ where all its elements are the lower bounds of q_{ij}, isn't surely the matrix that gives the lower stationary vector $\underline{\Pi}$. We have the same remark regarding the upper transition matrix $\overline{Q}$ where all its elements are the upper bounds of q_{ij}, isn't surely the matrix that gives the upper stationary vector $\overline{\Pi}$. In other words, there are some combinations of the transitions rates intervals, which can lead to a lower or higher matrix different from the ones chosen by the authors to find the solutions (vector of probabilities) which are outside the range of values proposed by the authors.

In fact, by applying the "Exact method", that we explain in the next section, we can understand that their assumption is not always correct. The main idea to get all possible combinations of the lower and upper bounds of q_{ij}, for each combination we have a transition matrix. With the transition matrix we find the vector Π then the availability of the system. We compare all the values of the availability, so that we could choose the matrix Q_l that belongs to $[\underline{Q}, \overline{Q}]$, and that gives the lower bound of $[\Pi]$, which corresponds to the

lowest vector among all the Πs and the lower bound of $[A]$. Q_l is given as follows:

$$Q_l = \begin{bmatrix} -0.83 & 0.32 & 0.32 & 0.19 \\ 730 & -1460.61 & 0 & 0.51 \\ 730 & 0 & -1460.61 & 0.51 \\ 0 & 730 & 730 & -2920 \end{bmatrix} \tag{4.12}$$

such that $[\underline{\Pi}_2, \overline{\Pi}_2]$ is the interval of Π obtained by the Exact method, with:

$$\underline{\Pi}_2 = \begin{bmatrix} 9.9816 \times 10^{-1} \\ 5.6745 \times 10^{-4} \\ 5.6745 \times 10^{-4} \\ 6.9782 \times 10^{-4} \end{bmatrix} \tag{4.13}$$

and another matrix Q_u that also belongs to $[\underline{Q}, \overline{Q}]$, that gives the upper bound of $[\Pi]$, which corresponds to the highest vector among all the Πs and the upper bound of $[A]$. Q_u is given as follows:

$$Q_u = \begin{bmatrix} -0.83 & 0.32 & 0.32 & 0.19 \\ 1460 & -1460.61 & 0 & 0.51 \\ 1460 & 0 & -730.51 & 0.51 \\ 0 & 730 & 730 & -2920 \end{bmatrix} \tag{4.14}$$

with:

$$\overline{\Pi}_2 = \begin{bmatrix} 9.994 \times 10^{-1} \\ 1.894 \times 10^{-4} \\ 3.787 \times 10^{-4} \\ 5.91 \times 10^{-5} \end{bmatrix} \tag{4.15}$$

The obtained availability, by doing the sum over the probabilities of the working states (in this example the sum of $[\pi_1]$, $[\pi_2]$ and $[\pi_3]$), is $[A_2] = [0.999302, 1.00005]$.
Table 4.1 show the availability obtained by the proposed method in the article and the availability obtained by the "Exact method". To show the efficiency of the "Exact method", we kept the obtained interval as it is, however in this case, we can make the intersection of the obtained interval with $[0, 1]$ since the probability belongs to this interval. Note that in this example and in the rest of the manuscript, the results are obtained after rounding calculations.

$Q_l \in [\underline{Q}, \overline{Q}]$ gives the lower stationary vector $\underline{\Pi}$, which will give the lower bound of the system's availability interval. Thus, $Q_u \in [\underline{Q}, \overline{Q}]$ gives the upper bound $\overline{\Pi}$, which will give the upper bound of the system's availability interval. Therefore, it is not like as it mentioned in Troffaes article that lower and upper bounds of $[\Pi]$ are obtained by using $\underline{Q}$ which all

The article's method	$[A_1] = [0.9990246, 1.00085]$
The Exact method	$[A_2] = [0.999302, 1.00005]$

Table 4.1: The imprecise availability obtained by the two methods

its elements are the lower bounds of the intervals and by $\overline{Q}$ which all its elements are the upper bounds of the intervals. Note that in the article, the authors did not take into account the condition that the sum of the probabilities is equal to 1. Later, in section 4.3 we detail our proposed method to solve the problem by taking into account the lower and the upper bounds of q_{ij}, with a method that will be called the "Exact" method.

4.3 The "Exact" method

Every MSS, formed from a number of components where each component could have a certain number of functioning states, has its own states. At each time *t*, the system has a probability $P_i(t)$ to be in the state *i*. To figure out, $A(t)$ the availability of the system at *t*, it depends on all the probabilities representing only the functioning states. For each system, there is an exact value of the availability that can be computed. In our case, we are dealing with uncertainties and the estimation of this exact value of the availability of the system is hard.

The failure rates and the repair rates are bounded by intervals, that is why when we want to find the transition matrix of the system, it will be also bounded by interval. The probability vector which its elements are the probability of the system to be in a certain state will be also bounded by interval. In 4.2.2, we presented a work that have been done in [110]. In this article, the authors state that the availability of the system will be bounded by interval, and that in order to find this interval, it will be enough to find the lower bound and the upper bound. Therefore, they proposed a method that helps to find the lower bound $\underline{\Pi}$ and the upper bound $\overline{\Pi}$ of the vector Π, where Π is the vector of probabilities of the system to be in the system's states when *t* tends to infinity. To find $\underline{\Pi}$ and $\overline{\Pi}$, we just need to find in order the lower bound of the transition matrix $\underline{Q}$ and the upper bound of the transition matrix $\overline{Q}$. However, $\underline{Q}$ and $\overline{Q}$ are not necessarily the transition matrices that gives the lower bound $\underline{\Pi}$ and the upper bound $\overline{\Pi}$.

In this section, we propose a method that we will call it "Exact" method, since it is the most guaranteed method that gives a narrow interval of availability of the system compared to other methods and that contains the real exact value of the availability that is impossible

to find it. The main idea of this method is simple; we need to take all the possible values of the tranisition matrix elements $[q_{ij}]$ either its lower bound or the upper bound (combinations of the bounds of $[\lambda]$ the failure rates intervals and $[\mu]$ the repair rates intervals of the systems components according to their position in the matrix). Which means all the possible k combinations. For each combination k, we find the corresponding transition matrix Q_k and we turn the problem into the precise case in order to solve eq. 4.1 to find the corresponding stationary vector Π_k. From the vector Π_k, we can calculate the corresponding availability A_k of the system for the combination k. The number of the possible combinations is 2^l, where *l* is the number of $[q_{ij}]$ the elements of the transition matrix $[Q]$, and $k \in \{1, \ldots, 2^l\}$. We compare all the obtained availabilities from all combinations, and we can choose the corresponding availability interval A_e, which is the availability of the system obtained by the "Exact method". Its bounds are the lowest availability (lower bound) and the highest availability (upper bound) between all the A_k.

Here a simple example to show how does this technique work:

Example: Considering a system formed by two components A and B. Each of these two components has two possible states: perfect functioning and total failure. Each component has the following failure and repair rates per hours:

$$\begin{cases} \lambda_A = [2 \times 10^{-3}, 4 \times 10^{-3}] \; ; \mu_A = [1.8 \times 10^{-2}, 3.5 \times 10^{-2}] \\ \lambda_B = [3 \times 10^{-3}, 4.5 \times 10^{-3}] \; ; \mu_B = [1.5 \times 10^{-2}, 3 \times 10^{-2}] \end{cases}$$

The system works if the two components work (see Figure 4.2). The system will have here 4 possible states:

- State 1: Components A and B work
- State 2: Component A works and B fails
- State 3: Component B works and A fails
- State 4: Components A and B fail

The Markov graph of the system states that shows the transitions between one state to another one is presented in Figure 4.3. Since the system is a series system that's mean that the system works unless all the components work, we can regroup the system's states into two states: Working state colored node (state 1) and failure state (states 2, 3 and 4).

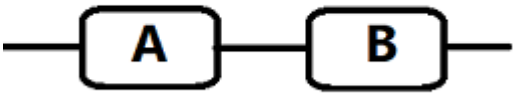


Figure 4.2: Series system of components A and B

The transition matrix elements q_{ij} can take either the lower or the upper bound of the interval $[q_{ij}]$ (the combination of the failure and the repair rates of A and B). For example, the first combination $k = 1$ is when all the q_{ij} take as values the lower bounds of their intervals. The second combination $k = 2$ is when q_{11} takes as value the upper bound of its interval and the rest take the lower bounds as values, and so on. In this case, we have $l = 16$ and $2^l = 2^{16}$ possibles precise transition matrices Q_k, with $k \in \{1, \ldots, 2^{16}\}$ in this form:

$$Q_k = \begin{bmatrix} -(\lambda_A + \lambda_B) & \lambda_A & \lambda_B & 0 \\ \mu_A & -(\mu_A + \lambda_B) & 0 & \lambda_B \\ \mu_B & 0 & -(\lambda_A + \mu_B) & \lambda_A \\ 0 & \mu_B & \mu_A & -(\mu_A + \mu_B) \end{bmatrix} \tag{4.16}$$

For each Q_k, we solve $\Pi_k.Q_k = 0$ to find the vector Π_k and the availability A_k. When comparing all the k availabilities we reform the interval of the availability of the system, where the lower bound has the smallest value between all the A_ks and the upper bound has the highest value between all of the A_ks.

In this example, the availability of the system is:

$$A = [0.5207, 0.9427].$$

With the corresponding Q_l the transition matrix that gives $\underline{A}$ and Q_u the transition matrix that gives $\overline{A}$ (P.S.: Q_l and Q_u are not the lower and upper transition matrix of Q).

$$Q_l = \begin{bmatrix} -0.005 & 0.002 & 0.003 & 1 \\ 0.018 & -0.038 & 0 & 1 \\ 0.015 & 0 & -0.032 & 1 \\ 0 & 0.015 & 0.018 & 1 \end{bmatrix}.$$

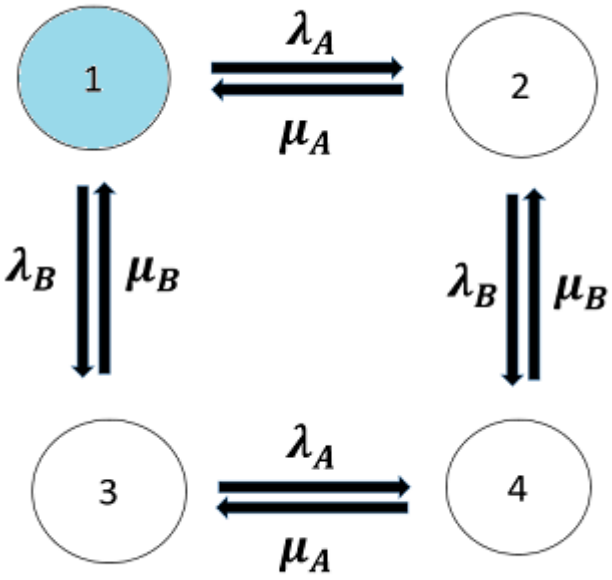


Figure 4.3: Markov graph of the series system

$$Q_u = \begin{bmatrix} -0.005 & 0.004 & 0.0045 & 1 \\ 0.035 & -0.038 & 0 & 1 \\ 0.03 & 0 & -0.032 & 1 \\ 0 & 0.015 & 0.018 & 1 \end{bmatrix}.$$

When uncertainties exist and when we use interval analysis to treat this uncertainties, this method is accurate and guarantee in the sense that the obtained interval will always contain the unknown precise value. However, we do not proceed like in [111] (cf. section 4.2.2). As we saw in the example presented in this section, that for a system of two binary components, we had 2^{16} possible cases. This means that the exact method would not be simple in terms of cost of time when having larger systems, in terms of number of components and number of states. If the system is larger, then the transition matrix will be large and the same for the number of cases which will be equal to 2^l. As a conclusion, the exact method will be time consuming and computing of the availability at each time will not be straightforward.

4.4 The proposed methodology

Our main problem is to calculate the availability of an MSS when t tends to ∞. We should take into account the uncertainties on the data (failure and repair rates). The first method presented in 4.2.2 cannot be used, since the authors found the lower bound of the stationary vector by taking the lower bound of the transition matrix and the upper bound of the sta-

tionary vector by taking the upper bound of the same matrix. We showed by detailing the calculation that there may exist another matrices that gives the upper and the lower bound of $[\Pi]$.

The exact method is the most efficient method to find the imprecise availability of an MSS, however when the system become much larger, the number of transitions will become very huge. In this case, we will have combinatorial explosion of the calculation. For this aim, we need a guaranteed method that helps us to calculate efficiently the imprecise availability of the MSS in order to guarantee that our solution include the true value of the availability. Our search will be for simple procedures for calculation and with a minimal cost of time.

We chose to model the uncertainties of data in terms of intervals, and we will use the imprecise Markov models and introducing the technique of contractors to compute the imprecise availability of an MSS.

In this section, we give the detailed steps of our methodology and to better demonstrated this methodology, we present the different steps through a small example.

4.4.1 The steps of the methodology

To calculate the availability of a complex MSS with multi-state components, we must follow and apply the steps presented in Figure 4.4. To summarize, we will follow the following steps:

1. Step 1: Define the Markov graph to present the states and all transitions between the states of the system.
2. Step 2: Construct the interval transition matrix of the system.
3. Step 3: Apply the technique of contraction on the system of equations $\Pi \cdot Q = 0$.
4. Step 4: Find Π and the availability of the system.

Before we start with the steps of the method, we need to list all the information about the system that we are dealing with. What kind of system we have and what is the structure of this system, and the number of his components and how many states does each component have? To this end, we determine the states of the components of the system and we define the corresponding interval failure rates λ and the interval repair rates μ of each component.

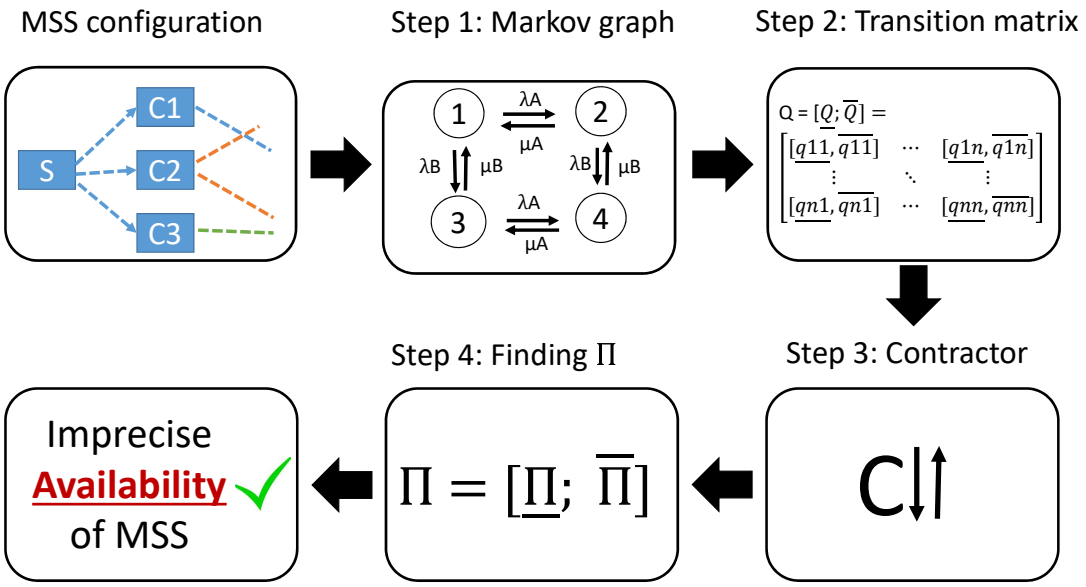


Figure 4.4: Steps to calculate the availability of the system

Then, we figure out the type of connection between one component's state and the other states. In this way, we can easily define the possible states of the system and understand the type of connections between them.

We should mention that when the components are independent, we can take each component apart and apply the steps of the methodology to compute its availability then we compute the availability of the system. The obtained interval will be more conservative than the availability's interval obtained when applying the methodology on the entire system. For illustration and of the different steps, we will take the following simple example:

Suppose a parallel system of two components A and B, which means that the system works when at least one of the two components works. Figure 4.5 presents the reliability block diagram of such system. For each component, we have two possible states (binary components): state 1 (perfect functioning) and state 2 (total failure). Thus, the whole system has 4 possible states, where the first three states present the functioning state of the parallel system and the last state is the failure state. The components have the following failure and repair rates per hour:

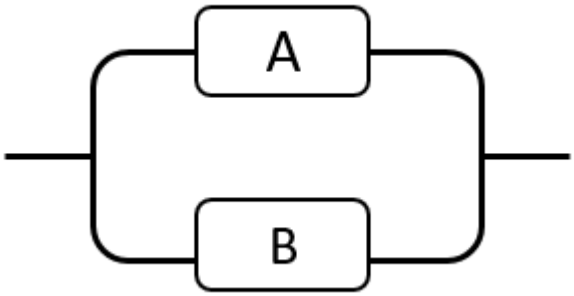


Figure 4.5: Bloc diagram of a parallel system of two components

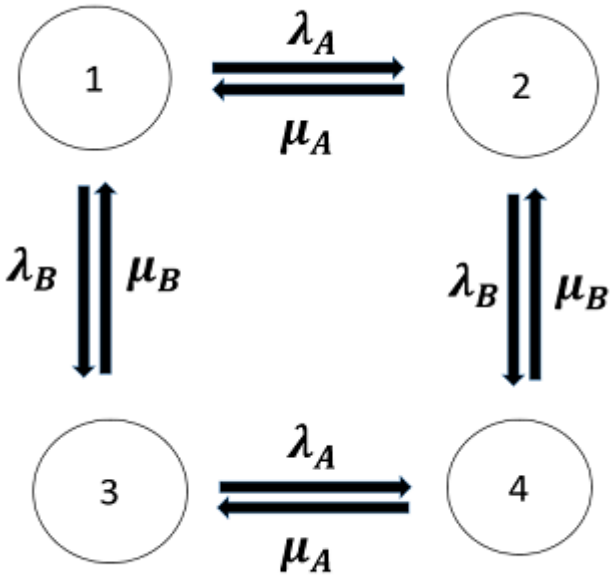


Figure 4.6: Markov chain of a system of two binary components

$$\begin{cases} \lambda_A = [3 \times 10^{-3}, 4 \times 10^{-3}] \; ; \mu_A = [2.8 \times 10^{-2}, 3.5 \times 10^{-2}] \\ \lambda_B = [3.5 \times 10^{-3}, 4.5 \times 10^{-3}] \; ; \mu_B = [2 \times 10^{-2}, 3 \times 10^{-2}] \end{cases}$$

Step 1

When knowing the system structure and the states of each component, we can easily find the states of the whole system. In the example above, we have 2 binary components. Thus, the system has 4 states: total functioning (state 1), degraded states (states 2 and 3), these three states are considering as functioning states since we have a parallel system and the last state is the total failure (state 4). Figure 4.6 shows the Markov graph of this system.

Step 2

When having the Markov graph either for the system or for each component, we pass to the second step. As in the precise case, we can figure out the transition matrix from the Markov graph, the only difference here is that we are dealing with uncertainties in terms of interval. We define the interval transition matrix $[Q]$ where each term of the matrix $q_{ij} = [\underline{q_{ij}}, \overline{q_{ij}}]$, is an interval of failure rates $\lambda_i = [\underline{\lambda_i}, \overline{\lambda_i}]$ and interval of repair rates $\mu_i = [\underline{\mu_i}, \overline{\mu_i}]$ of dimension $(n \times n)$.

We should mention that each term of the transition matrix represents the transition rate from one state to the other one. The transition matrix of our example is:

$$[Q] = [\underline{Q}, \overline{Q}] = \begin{bmatrix} [\underline{q_{11}}, \overline{q_{11}}] & [\underline{q_{12}}, \overline{q_{12}}] & [\underline{q_{13}}, \overline{q_{13}}] & [\underline{q_{14}}, \overline{q_{14}}] \\ . & . & . & . \\ . & . & . & . \\ [\underline{q_{41}}, \overline{q_{41}}] & [\underline{q_{42}}, \overline{q_{42}}] & [\underline{q_{43}}, \overline{q_{43}}] & [\underline{q_{44}}, \overline{q_{44}}] \end{bmatrix} \tag{4.17}$$

In our example, we will have $[Q]$ as:

$$\begin{bmatrix} [-(\overline{\lambda_A} + \overline{\lambda_B}), -(\underline{\lambda_A} + \underline{\lambda_B})] & [\underline{\lambda_A}, \overline{\lambda_A}] & [\underline{\lambda_B}, \overline{\lambda_B}] & 0 \\ [\underline{\mu_A}, \overline{\mu_A}] & [-(\overline{\mu_A} + \overline{\lambda_B}), -(\underline{\mu_A} + \underline{\lambda_B})] & 0 & [\underline{\lambda_B}, \overline{\lambda_B}] \\ [\underline{\mu_B}, \overline{\mu_B}] & 0 & [-(\overline{\lambda_A} + \overline{\mu_B}), -(\underline{\lambda_A} + \underline{\mu_B})] & [\underline{\lambda_A}, \overline{\lambda_A}] \\ 0 & [\underline{\mu_B}, \overline{\mu_B}] & [\underline{\mu_A}, \overline{\mu_A}] & [-(\overline{\mu_A} + \overline{\mu_B}), -(\underline{\mu_A} + \underline{\mu_B})] \end{bmatrix}$$

By replacing the values of failure rates and repair rate, we obtain:

$$\begin{bmatrix} [-8.5 \times 10^{-3}, -6.5 \times 10^{-3}] & [3 \times 10^{-3}, 4 \times 10^{-3}] & [3.5 \times 10^{-3}, 4.5 \times 10^{-3}] & 0 \\ [2.8 \times 10^{-2}, 3.5 \times 10^{-2}] & [-3.95 \times 10^{-2}, -3.15 \times 10^{-2}] & 0 & [3.5 \times 10^{-3}, 4.5 \times 10^{-3}] \\ [2 \times 10^{-2}, 3 \times 10^{-2}] & 0 & [-3.4 \times 10^{-2}, -2.3 \times 10^{-2}] & [3 \times 10^{-3}, 4 \times 10^{-3}] \\ 0 & [2 \times 10^{-2}, 3 \times 10^{-2}] & [2.8 \times 10^{-2}, 3.5 \times 10^{-2}] & [-6.5 \times 10^{-2}, -4.8 \times 10^{-2}] \end{bmatrix}$$

Step 3

After that we have our transition matrix $[Q]$, we start with the next step. To determine the availability of the system, we must have the probabilities of being in each state of the system, and since we want to find the availability when the time *t* tends to ∞, we need to find the steady probabilities of being in each state of the system:

$$[\Pi] = [\underline{\Pi}, \overline{\Pi}] = [[\underline{\pi_1}, \overline{\pi_1}] \dots [\underline{\pi_n}, \overline{\pi_n}]].$$

For this aim, as this vector is unknown for us, we will take an initial supposed vector $[\Pi]$ as:

$$[\Pi] = [[0,1][0,1]\dots[0,1]] \tag{4.18}$$

where $[\Pi]$ present all the initial intervals of steady probability of the system to be in each state. To compute the intervals of this vector, we will use the technique of contractors, that's why we apply eq.4.1 and we obtain a system of *n* equations with the last equation representing the fact that:

$$\sum_{i=1}^{n} \pi_i = 1 \text{ which means the sum of all the probabilities is equal to 1} \tag{4.19}$$

By doing the operations of the forward-backward propagation contractor, we will be able to contract our steady probabilities into smaller intervals that contains the real unknown exact values.

In the previous example, our system of equations is:

$$\begin{cases} [q_{11}][\pi_1] + [q_{21}][\pi_2] + [q_{31}][\pi_3] = 0 \Rightarrow (1) \\ [q_{12}][\pi_1] + [q_{22}][\pi_2] + [q_{41}][\pi_4] = 0 \Rightarrow (2) \\ [q_{13}][\pi_1] + [q_{33}][\pi_3] + [q_{43}][\pi_4] = 0 \Rightarrow (3) \\ [q_{24}][\pi_2] + [q_{34}][\pi_3] + [q_{44}][\pi_4] = 0 \Rightarrow (4) \\ [\pi_1] + [\pi_2] + [\pi_3] + [\pi_4] = 1 \Rightarrow (5) \end{cases} \tag{4.20}$$

We apply the Forward-Backward, on the intervals $[\pi_i] = \left[\underline{\pi_i}, \overline{\pi_i}\right]$, to reduce each $[\pi_i]$ as much as it is possible which means reducing the elements of vector Π.

We start with the first contraction, each constraint (equation) in 4.20 has the form:

$$f_i(\pi_1, \dots, \pi_{n_k}) = 0,$$

where f_i can be decomposed into a sequence of operations involving elementary operators such as $(+, -, *, /)$. We decompose this constraint into primitive constraints [65]. A primitive constraint is a constraint involving a single operator. For instance the first constraint in 4.20, can be decomposed into the following primitive constraints:

$$\begin{cases} [a_1] = [q_{11}][\pi_1] = [-8.5 \times 10^{-3}, -6.5 \times 10^{-3}] * [0,1] = [-8.5 \times 10^{-3}, 0], \\ [a_2] = [q_{21}][\pi_2] = [2.8 \times 10^{-2}, 3.5 \times 10^{-2}] * [0,1] = [0, 3.5 \times 10^{-2}], \\ [a_3] = [q_{31}][\pi_3] = [2 \times 10^{-2}, 3 \times 10^{-2}] * [0,1] = [0, 3 \times 10^{-2}], \\ [y] = [a_1] + [a_2] + [a_3] = [-8.5 \times 10^{-3}, 6.5 \times 10^{-2}] \end{cases} \tag{4.21}$$

When having the new interval [y], we find the intersection with 0 because the main constraint $y = 0$:

$$[y] \cap 0 = [-8.5 \times 10^{-3}, 6.5 \times 10^{-2}] \cap [0,0] = [0,0].$$

With this final value of [y], we make the backward propagation, by re-finding the three intervals $[a_1], [a_2]$ and $[a_3]$, so at the end we can reconstruct the needed intervals $[\pi_1], [\pi_2]$ and $[\pi_3]$, as shown in 4.22:

$$\begin{cases} [a_1] = ([y] - [a_2] - [a_3]) \cap [a_1] \\ [a_2] = ([y] - [a_1] - [a_3]) \cap [a_2] \\ [a_3] = ([y] - [a_1] - [a_2]) \cap [a_3] \\ [\pi_1] = ([a_1]/q_{11}) \cap [\pi_1] \\ [\pi_2] = ([a_2]/q_{21}) \cap [\pi_2] \\ [\pi_3] = ([a_3]/q_{31}) \cap [\pi_3] \end{cases} \tag{4.22}$$

The obtained intervals are:

$$\begin{cases} [a_1] = [-6.5 \times 10^{-2}, 0] \cap [-8.5 \times 10^{-3}, 0] = [-8.5 \times 10^{-3}, 0] \\ [a_2] = [-3 \times 10^{-2}, 8.5 \times 10^{-3}] \cap [0, 3.5 \times 10^{-2}] = [0, 8.5 \times 10^{-3}] \\ [a_3] = [-3.5 \times 10^{-2}, 8.5 \times 10^{-3}] \cap [0, 3 \times 10^{-2}] = [0, 8.5 \times 10^{-3}] \\ [\pi_1] = [0, 1.30769] \cap [0,1] = [0,1] \\ [\pi_2] = [0, 0.3035714] \cap [0,1] = [0, 0.3035714] \\ [\pi_3] = [0, 0.425] \cap [0,1] = [0, 0.425] \end{cases} \tag{4.23}$$

After applying the constraint number 1, we obtained the vector Π in the form:

$$[\Pi_1] = \Big[[0,1][0, 0.3035714][0, 0.425][0,1]\Big] \tag{4.24}$$

We continue the technique of contraction with the constraint number 2. By following the same steps, we have the primitive constraints:

$$\begin{cases} [b_1] = [q_{12}][\pi_1] = [3 \times 10^{-3}, 4 \times 10^{-3}] * [0, 1] = [0, 4 \times 10^{-3}], \\ [b_2] = [q_{22}][\pi_2] = [-3.95 \times 10^{-2}, -3.15 \times 10^{-2}] * [0, 0.3035714] = [-1.199107 \times 10^{-2}, 0], \\ [b_4] = [q_{42}][\pi_4] = [2 \times 10^{-2}, 3 \times 10^{-2}] * [0, 1] = [0, 3 \times 10^{-2}], \\ [y] = [b_1] + [b_2] + [b_4] = [-1.199107 \times 10^{-2}, 7 \times 10^{-2}] \end{cases} \tag{4.25}$$

The intersection of [y] with $[0, 0]$, yeld to $[0, 0]$. Now we repeat the same operations backward to re-find $[b_1]$, $[b_2]$ and $[b_4]$, then the newest version of $[\pi_1]$, $[\pi_2]$ and $[\pi_4]$. We find:

$$\begin{cases} [b_1] = ([y] - [b_2] - [b_4]) \cap [b_1] \\ [b_2] = ([y] - [b_1] - [b_4]) \cap [b_2] \\ [b_4] = ([y] - [b_1] - [b_2]) \cap [b_4] \\ [\pi_1] = ([b_1]/q_{12}) \cap [\pi_1] \\ [\pi_2] = ([b_2]/q_{22}) \cap [\pi_2] \\ [\pi_4] = ([b_4]/q_{42}) \cap [\pi_4] \end{cases} \tag{4.26}$$

With:

$$\begin{cases} [b_1] = [-3 \times 10^{-2}, 1.199107 \times 10^{-2}] \cap [0, 4 \times 10^{-3}] = [0, 4 \times 10^{-3}] \\ [b_2] = [-3.4 \times 10^{-2}, 0] \cap [-1.199107 \times 10^{-2}, 0] = [-1.199107 \times 10^{-2}, 0] \\ [b_4] = [-4 \times 10^{-3}, 1.199107 \times 10^{-2}] \cap [0, 3 \times 10^{-2}] = [0, 1.199107 \times 10^{-2}] \\ [\pi_1] = [0, 1.33333] \cap [0, 1] = [0, 1] \\ [\pi_2] = [0, 0.3807] \cap [0, 0.3035714] = [0, 0.3035714] \\ [\pi_4] = [0, 0.05996] \cap [0, 1] = [0, 0.05996] \end{cases} \tag{4.27}$$

Where the newest vector of probabilities is:

$$[\Pi_2] = \Big[[0, 1][0, 0.3035714][0, 0.425][0, 0.05996]\Big] \tag{4.28}$$

Now, we take the third constraint, then the fourth constraint to end with the final one, and with each constraint we apply the same previous steps. Table 4.2 represents the obtained probability vector Π after using the five constraints and by applying the technique of contractors only one time.

Constraint number	Probability vector Π
1	$[0,1][0,0.3035714][0,0.425][0,1]$
2	$[0,1][0,0.3035714][0,0.425][0,0.05996]$
3	$[0,1][0,0.3035714][0,0.032084][0,0.5996]$
4	$[0,1][0,0.3035714][0,0.425][0,0.0065544]$
5	$[0.207552,1][0,0.3035714][0,0.425][0,0.0638764]$

Table 4.2: The obtained probability vector after each constraint

Number of contraction *k*	Probability vector Π
1	$[0.207,1][0,0.3035][0,0.425][0,0.0638]$
2	$[0.4968,0.9599][0.0161,0.182][0.022,0.2806][0.0018,0.0404]$
3	$[0.571,0.9013][0.0396,0.1554][0.0542,0.2389][0.0046,0.0344]$
6	$[0.6252,0.8719][0.0513,0.1359][0.0707,0.2086][0.006,0.0301]$
10	$[0.6288,0.8698][0.0521,0.1346][0.0718,0.2066][0.0061,0.0298]$

Table 4.3: The obtained probability vector after *k* contraction

We keep repeating the contractions on all the equations and several times, until π the elements of the vector Π converge. That is how we obtain the final version of the interval vector Π. Table 4.3 presents the vector Π after *k* times of contraction.
Figure 4.7 shows how the probability interval for each state contracts at each time.

Step 4

We calculate the imprecise availability of the system A_s by doing the sum of π_i over all the working states.

$$A_s = \sum_{i, \text{ over the working states}} [\pi_i] \tag{4.29}$$

In the example, we have a parallel system of 2 components, π_1, π_2, π_3 are the probabilities of the system to be in a working state. π_4 is the probability to be in the failure state.
From the last version of $[\Pi]$ obtained after 10 contractions (when each $[\pi_i]$ converges to a certain interval), we compute the availability of the system:

$$\begin{cases} [\pi_1] + [\pi_2] + [\pi_3] = [0.7529529, 1.21107924] \\ A_s = [0.7529529, 1.21107924] \end{cases} \tag{4.30}$$

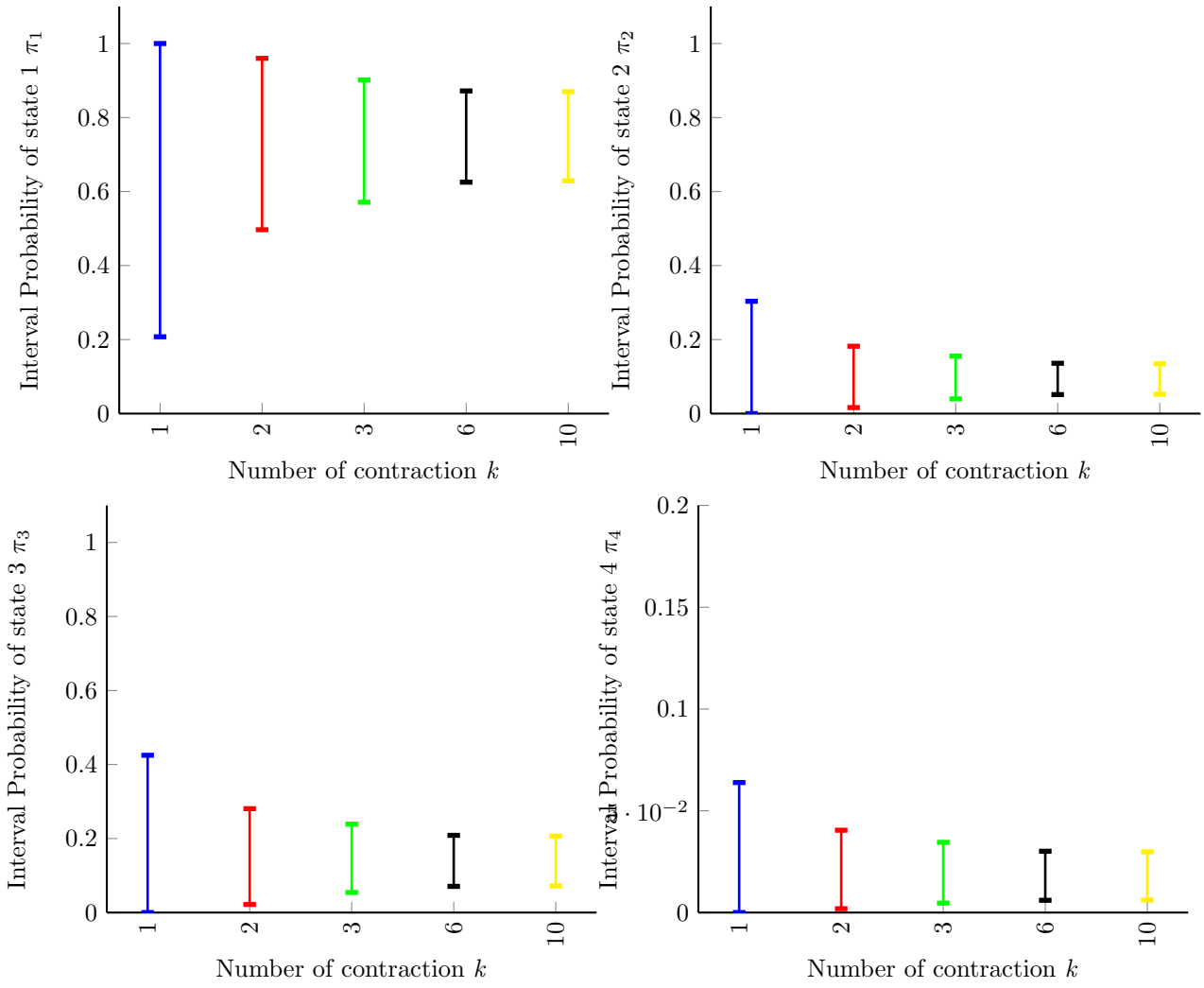


Figure 4.7: The probability interval for each state after *k* contractions

Note that we can find the availability of the system when its components are independent, by taking each component apart and applying the same steps mentioned in this section. By doing the steps for each component, we will find the corresponding $[\Pi_j]$ of the component j, then we can compute the availability of each component $[A_j]$ separately. When having the availability of all the components, we figure out the relation between all the $[A_j]$ and we could calculate the imprecise availability of the system $[A_s]$. In the example, we have a parallel system which means that the relation between the availability of components is:

$$A_s = 1 - (1 - A_1)(1 - A_2) \tag{4.31}$$

To verify if the result obtained by the technique of contractors is accurate, we will apply the "Exact" method and compare the obtained availability by the technique of contractors to

the availability obtained by the "Exact" method. We recall that the main idea of the "Exact" method is to take all the possible transition matrices by doing a combination between the lower bounds and upper bounds of $[q_{ij}]$ (by taking the lower and upper bounds of the failure rates interval and the repair rates interval). For each combination we obtain a transition matrix. We solve the system from eq 4.1 and find the corresponding $[\Pi_k]$ and the availability $[A_k]$ where we could have 2^l possibilities where l is the number of $[q_{ij}]$ the elements of the transition matrix $[Q]$, and $k = 1, \ldots, 2^l$, after that we compare all the obtained availabilities from all combinations, and we can choose the corresponding availability interval $[A_E]$ where its bounds are the lowest availability (lower bound) and the highest availability (upper bound).

When we retake the previous example, we obtain the lower bound of the steady probabilities obtained by the Exact method is:

$$\underline{\Pi}_E = \begin{bmatrix} 0.666963 \\ 0.098112 \\ 0.146102 \\ 0.088821 \end{bmatrix}$$

and the upper bound of the steady probabilities obtained by the Exact method is:

$$\overline{\Pi}_E = \begin{bmatrix} 0.837987 \\ 0.060367 \\ 0.111134 \\ 0.009490 \end{bmatrix}$$

The availability by the Exact method is:

$$[A_E] = [0.91117, 1.00949]$$

In addition, we can verify the results, by taking the center of each interval elements of the transition matrix $[q_{ij}]$, we will get one transition matrix formed from all the midpoint values of the data, we solve eq 4.1 to get the precise midpoint vector $mid([\Pi])$, where we could finally calculate the precise midpoint availability $mid([A])$, which it has to belong to the interval availability obtained by the technique of contractors $[A]$ and to the interval availability obtained by the "Exact" method $[A_E]$. We will call this method the "Precise method", with $mid([A]) = A_P$.
When we retake the previous example, we obtain $A_P = 0.986206897$.
Table 4.4 presents the obtained results by the three methods.

The technique of contractors	The Exact method	The Precise method
$A_s = [0.75295, 1.21107]$	$[A_E] = [0.91117, 1.00949]$	$A_P = 0.98620$

Table 4.4: The obtained system's availability by different methods

Figure 4.4 illustrates the steps of computing the availability of the system by using the technique of contractors.

4.5 Metrics for validation of the methodology

4.5.1 Guaranteed Index (GI)

In general, the results provided by the forward-backward propagation methods and the contractors are guaranteed, which means that from correct assumptions, interval contraction will provide correct conclusions. The results are guaranteed in any case.

More specifically, in availability analysis, we consider a guaranteed availability result, a result obtained when we are sure that the "real" availability belongs to the obtained availability interval. To do so, and in order to quantify and validate the efficiency of our method, we have considered that a guaranteed result i.e. an interval which contains the exact interval availability. We recall that the exact interval availability is obtained when considering all the combinations the transition matrix elements q_{ij} (upper and lower failure and repair rates) and we compute the optimum interval from all the obtained intervals.

Thus, we introduce, the "Guaranteed Index (GI)" which is a binary variable equal to one if the obtained availability interval contains the exact interval availability, and 0 if it is not the case.

4.5.2 Conservatism Rate (CR)

With regard to conservatism, we define a conservatism rate of the obtained availability using a method as the ratio of the difference between the width of the interval of this availability obtained by our proposed method and the width of the one obtained by the exact method, and the width of the availability obtained by our proposed method.

Formally, the Conservatism Rate (CR) will be given by:

$$CR = \frac{w([A_{obtained}]) - w(A_E)}{w(A_{obtained})} \tag{4.32}$$

where $w([A]) = \overline{A} - \underline{A}$.

4.6 Case studies

To understand and validate the proposed methodology, we will propose two cases of study where we have an MSS and apply the methodology to calculate the imprecise availability. When having the imprecise availability, we will compare the results to the ones obtained by two principal methodologies (IUGF [66] and BUGF [28]) presented in chapter 3. The first case of study is used to prove the accuracy of our method comparing to other methods. The second one, is used to verify that the method could handle efficiently the cases when the systems are complex with a large number of components and states.

4.6.1 Case study 1

In this section, we use the example presented by Soroudi et al. [105]. Our aim is to compute the availability of a flow transmission system presented in Figure 4.8 and made of three pipes. The flow is transmitted from left to right, and the performances of the pipe are measured by their transmission capacity (tons per minute).

It is supposed that components 1 and 2 have three states: a state of total failure corresponding to a capacity of 0, a state of full capacity, and a state of partial failure. Only the third component has two states of functioning: a state of total failure, and a state of full capacity. All the performance's level of the component's states are precise. We want to calculate the availability of the system by using Markov chain and we will compare it to the availability obtained by the IUGF proposed in [66] and the availability obtained by the BUGF proposed in [28].

In this case study, we consider that both failure and repair rates are given by intervals. Table 4.5 presents the failure and repair rates of the three components.

Moreover, each of the components 1 and 2 have three possible states, while component 3 has only two states, hence the whole system has at most eighteen $(3 \times 3 \times 2)$ possible states degrading from the total functional state to the failure state as shown in Figure 4.9.

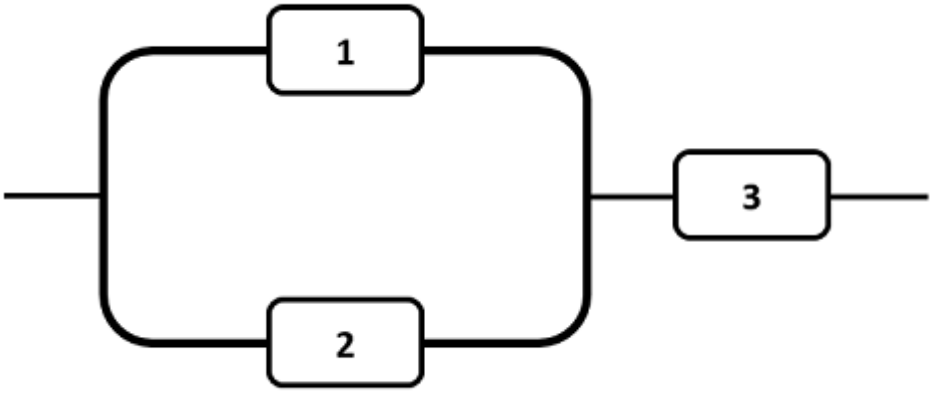


Figure 4.8: Case study 1: Flow transmission system

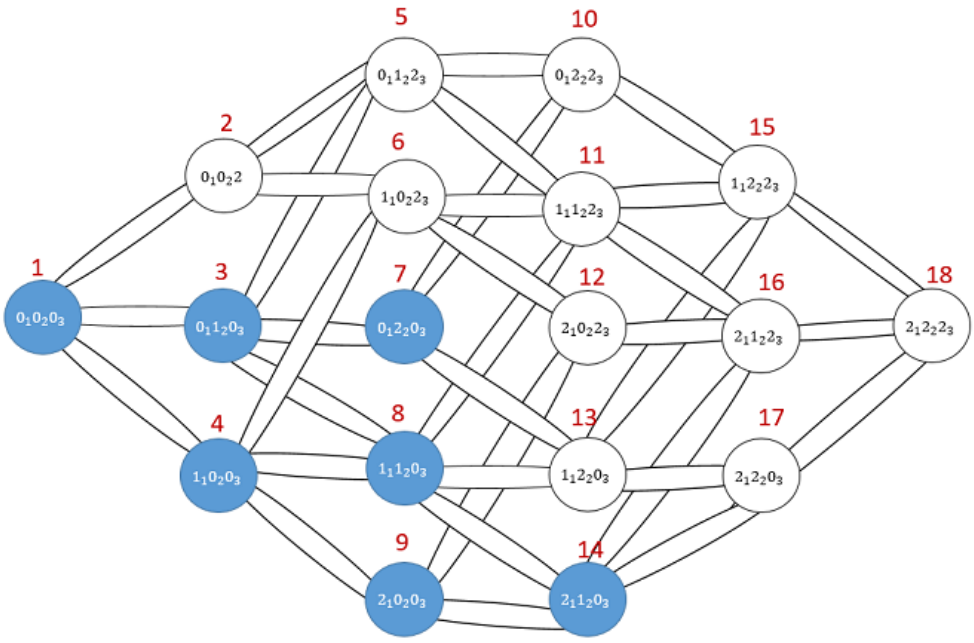


Figure 4.9: Case study1: Markov chain for 18 states

	G_1	G_2	G_3
State 1	$g_3^1 = 1.5$ (Completely successful)	$g_3^2 = 2$ (Completely successful)	$g_2^3 = 4$ (Completely successful)
State 2	$g_2^1 = 1$ (Degraded successful)	$g_2^2 = 1.5$ (Degraded successful)	$g_1^3 = 0$ (Total failure)
State 3	$g_1^1 = 0$ (Total failure)	$g_1^2 = 0$ (Total failure)	–
Failure rates	$\lambda_{3,1}^1 = [10^{-5}, 3 \times 10^{-4}]\ h^{-1}$ $\lambda_{2,1}^1 = [4 \times 10^{-5}, 5 \times 10^{-4}]\ h^{-1}$	$\lambda_{3,1}^2 = [2 \times 10^{-5}, 6 \times 10^{-4}]\ h^{-1}$ $\lambda_{2,1}^2 = [3 \times 10^{-5}, 4 \times 10^{-4}]\ h^{-1}$	$\lambda_{2,1}^3 = [10^{-5}, 4 \times 10^{-4}]\ h^{-1}$
Repair rates	$\mu_{2,1}^1 = [2 \times 10^{-2}, 5 \times 10^{-2}]\ h^{-1}$ $\mu_{1,1}^1 = [4 \times 10^{-2}, 8 \times 10^{-2}]\ h^{-1}$	$\mu_{2,1}^2 = [3 \times 10^{-2}, 6 \times 10^{-2}]\ h^{-1}$ $\mu_{1,1}^2 = [3 \times 10^{-2}, 7 \times 10^{-2}]\ h^{-1}$	$\mu_{1,1}^3 = [5 \times 10^{-2}, 9 \times 10^{-2}]\ h^{-1}$

Table 4.5: Transition rates and states for each component

In the figure: 0_j means that component *j* is in the completely working state, 1_j means that component *j* is in the partial working state and 2_j means that component *j* is in the completely failure state. In each state, the performance level *g* is calculated by taking into account the performance level of each component. For example, state 1 denoted by $0_1 0_2 0_3$ refers to the state where all three components are completely working. To obtain the performance level of the state: Components 1 and 2 are placed in parallel so the performance level resulting of them is the sum of the two performances which means $2 + 1.5 = 3.5$. Component 3 is placed in series, the total performance level *g* is the minimum, i.e. $g = min(3.5, 4) = 3.5$. In this example, we are studying the availability of the system for a demand level $w = 1.5$, we will consider the working states when the total performance level is greater than or equal to *w*. In Figure 4.9, the colored states present the working states.

First, we calculate the corresponding availability of the system using IUGF and BUGF, results are grouped in Table 4.6. Then we will compare these values to those obtained by our methodology. We first use the Forward-backward propagation technique to solve the system of equations by contracting the intervals. The system of equations 4.1 is equal to zero (18 equations) plus one last equation that is the sum of each steady availability term π is equal to one. By following the steps of the methodology, we find our imprecise availability. Our obtained results are shown in table 4.6. The result of the exact method is also given in Table 4.6 and in Figure 4.10.

Table 4.6 shows that the accuracy offered by the FBP contraction technique is close to that obtained when using the exact method; the interval availability by the technique of

Exact method	Contraction technique	BUGF	IUGF
$[0.9809, 1.0323]$	$[0.9551, 1.0360]$	[0.9550 , 1.0361]	[0.9137 , 1.0900]

Table 4.6: Availability whhen t tends to ∞ for each method

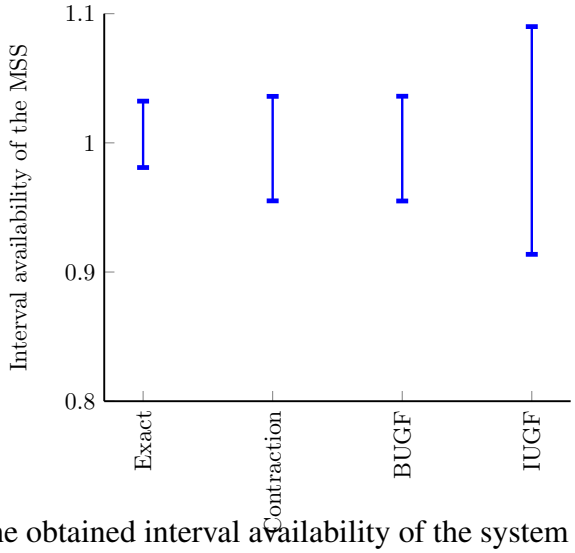


Figure 4.10: The obtained interval availability of the system by each method

contractors method is also more conservative than the other two methods (BUGF and IUGF), since the interval obtained by the contraction technique contains the interval obtained by the exact method and it is smaller than the intervals obtained by the IUGF method and very close the one obtained by the BUGF method.
In this case, the **GI** of this example is one since the obtained interval by using the contraction technique contains the interval obtained by the exact method. Also, the **CR** of this example is:

$$CR_1 = \frac{0.0809 - 0.0514}{0.0809} = 0.364 \tag{4.33}$$

Therefore, the FBP contraction method turns out as an efficient technique for availability assessment since it offers accurate results and it is more simple especially when handling complex systems.

4.6.2 Case study 2

After that we showed that the method is efficient and more accurate than the other methods to calculate the availability, now we will show that the proposed methodology remains efficient also when the system has a large number of states.

To illustrate this fact, we will apply it on the following case of study: we aim to compute the availability of the system presented in Figure 4.11. The system is composed of

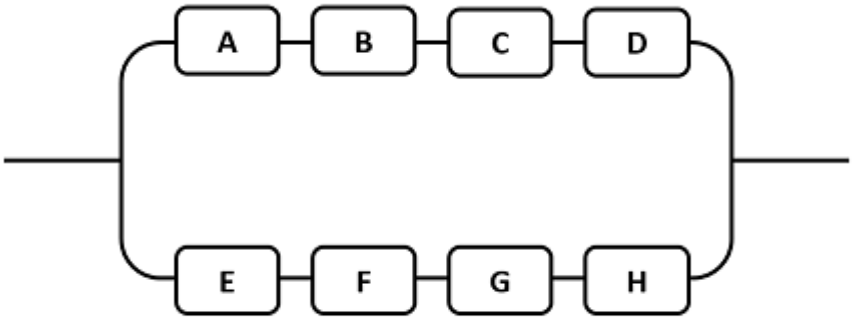


Figure 4.11: Case study 2: MSS composed of 8 components

8 components : A, B, C, D, E, F, G and H. The transmission of the system is from left to right. We assume that components B, C, E and H are binary components, i.e., they have only two possible states: a state of total failure, and a state of full functioning. The components A, F and G have three possible states: a state of total failure, a state of full functioning, and a state of partial failure. The component D has 4 possible degraded states: a state of total failure, a state of full functioning, a state of degraded functioning of type 1, and a state of degraded functioning of type 2. Thus, the total number of states for the system is $4 \times 3 \times 3 \times 3 \times 2 \times 2 \times 2 \times 2 = 1728$. The values of repair rates and failure rates of each component are presented in Figure 4.7, where $\lambda_{(k,i)}$ and $\mu_{(k,i)}$represent the failure rate i and repair rate *i* of component *k*.

Since in this example we consider that the components are independent, for each component, we will construct its Markov graph (Figure 4.12), in order to determine its interval transition matrix. Then, we will apply eq 4.1 to find $\Pi_{(k)}$ for each component, with $k = A, \ldots, H$. Finally, we compute the availability of each component by computing the sum of $\Pi_{(k,i)}$ over all the working states. In this example, the availability of the entire system A_{system} is presented in the following equation:

$$A_{system} = 1 - (1 - A_A A_B A_C A_D)(1 - A_E A_F A_G A_H) \tag{4.34}$$

To find $\Pi_{(k)}$ for each component, we will apply the technique of contractors. We propose to compare the obtained availability with the availability of the system obtained by the exact method and by the precise one. To simplify the comparison, we chose to write the results in terms of the unavailability $(1 - availability)$. All the results are presented in Table 4.8 and in Figure 4.13.

In this example the obtained availability interval by the contraction technique contains

Component	Failure rate	Repair rate
A	$\lambda_{1,1} = [3 \times 10^{-5}, 4.5 \times 10^{-5}]\ h^{-1}$ $\lambda_{1,2} = [2.6 \times 10^{-5}, 7 \times 10^{-5}]\ h^{-1}$	$\mu_{1,1} = [4.1 \times 10^{-1}, 7 \times 10^{-1}]\ h^{-1}$ $\mu_{1,2} = [3 \times 10^{-1}, 6.2 \times 10^{-1}]\ h^{-1}$
B	$\lambda_2 = [2.6 \times 10^{-5}, 3.1 \times 10^{-5}]\ h^{-1}$	$\mu_2 = [1.6 \times 10^{-1}, 4 \times 10^{-1}]\ h^{-1}$
C	$\lambda_3 = [3.5 \times 10^{-5}, 4.6 \times 10^{-5}]\ h^{-1}$	$\mu_3 = [2.1 \times 10^{-1}, 3 \times 10^{-1}]\ h^{-1}$
D	$\lambda_{4,1} = [2.6 \times 10^{-5}, 3.5 \times 10^{-5}]\ h^{-1}$ $\lambda_{4,2} = [2.9 \times 10^{-5}, 3.7 \times 10^{-5}]\ h^{-1}$ $\lambda_{4,3} = [2.7 \times 10^{-5}, 3.4 \times 10^{-5}]\ h^{-1}$	$\mu_{4,1} = [3.1 \times 10^{-1}, 3.5 \times 10^{-1}]\ h^{-1}$ $\mu_{4,2} = [2.9 \times 10^{-1}, 3.3 \times 10^{-1}]\ h^{-1}$ $\mu_{4,3} = [3 \times 10^{-1}, 3.5 \times 10^{-1}]\ h^{-1}$
E	$\lambda_5 = [3 \times 10^{-5}, 4.1 \times 10^{-5}]\ h^{-1}$	$\mu_5 = [2.5 \times 10^{-1}, 3 \times 10^{-1}]\ h^{-1}$
F	$\lambda_{6,1} = [3.1 \times 10^{-5}, 5.6 \times 10^{-5}]\ h^{-1}$ $\lambda_{6,2} = [2.9 \times 10^{-5}, 4.6 \times 10^{-5}]\ h^{-1}$	$\mu_{6,1} = [2 \times 10^{-1}, 2.4 \times 10^{-1}]\ h^{-1}$ $\mu_{6,2} = [2.6 \times 10^{-1}, 3 \times 10^{-1}]\ h^{-1}$
G	$\lambda_{7,1} = [4.1 \times 10^{-5}, 4.7 \times 10^{-5}]\ h^{-1}$ $\lambda_{7,2} = [2.9 \times 10^{-5}, 3.2 \times 10^{-5}]\ h^{-1}$	$\mu_{7,1} = [2.3 \times 10^{-1}, 2.9 \times 10^{-1}]\ h^{-1}$ $\mu_{7,2} = [2.7 \times 10^{-1}, 3.1 \times 10^{-1}]\ h^{-1}$
H	$\lambda_8 = [4.1 \times 10^{-5}, 5 \times 10^{-5}]\ h^{-1}$	$\mu_8 = [2.7 \times 10^{-1}, 3 \times 10^{-1}]\ h^{-1}$

Table 4.7: Failure and repair rates from a state to another for each component

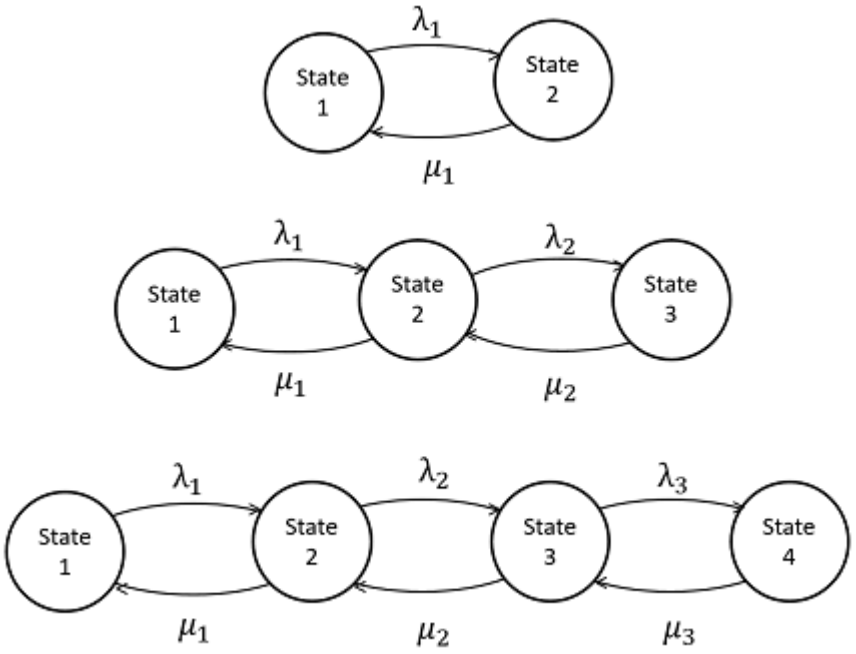


Figure 4.12: Case study 2: The corresponding Markov graph for each component

Exact method	Contraction technique	Precise method
$[3.318 \times 10^{-8}, 2.987 \times 10^{-7}]$	$[8.508 \times 10^{-9}, 5.310 \times 10^{-7}]$	8.346×10^{-8}

Table 4.8: The unavailability of the system using different methods

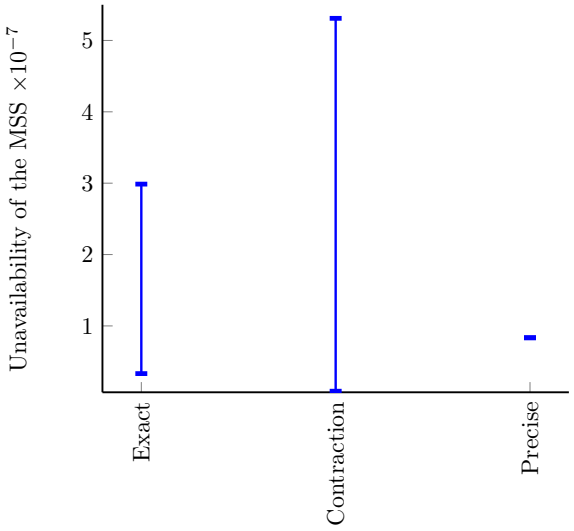


Figure 4.13: The obtained unavailability of the system by each method

the interval obtained by the exact method, the **GI** also in this example is one. When we calculate the **CR** for this case, we will have:

$$CR_2 = \frac{5.224 \times 10^{-7} - 2.655 \times 10^{-7}}{5.224 \times 10^{-7}} = 0.491 \tag{4.35}$$

Table 4.8 shows that the interval obtained by the contraction technique contains the interval obtained by the exact method and the precise availability belongs to the two availability intervals. Therefore, the FBP contraction method turns out as an efficient technique for availability assessment since it offers accurate results when having larger systems. However, $CR_2 > CR_1$ since the second system is larger than the first one with a big number of components and a big number of states.

4.6.3 Discussion of results and conclusion

In this chapter, we started with explaining what is a complex system, and that there is a difference between a binary system and an MSS. In our work, we are searching for the availability of a complex MSS. This problematic is not simple as it looks like, especially in presence of epistemic uncertainties (interval values of failure rates and repair rates). We want to estimate an imprecise availability of a complex MSS.

To find this imprecise availability, we chose to use imprecise Markovian model. Section 4.2 is devoted to explain the imprecise Markovian model in dependability. To find the imprecise availability of the system, we need to find the steady probability vector Π that presents the steady probabilities π to be in each state of the system. For that reason we need

to solve the system of equations in eq. 4.1. In the same section, we presented the only work that we know that is looking for the same aim. We recalled the proposed method in this work and we detailed an example that is presented in the article.

Later, we explained the "Exact method" which is a method based on finding all the possible precise cases and all the possible precise availabilities, and then we reconstruct the imprecise availability interval.

After, recalling the two methods IUGF and BUGF, we presented our proposed methodology, which consists the use of the technique of contractors in interval analysis to solve eq. 4.1 and find the imprecise availability of the system. To verify the methodology, two cases of study we proposed, the first is by comparing the result of our methodology with the results obtained by the "Exact method" and the two methods introduced in chapter 3, the IUGF and the BUGF. The second case, the obtained result of our methodology is compared to the results obtained by the "Exact method" and the "Precise method, which is the method that calculate the centric availability.

The results obtained from the first numerical application, by taking a simple case of multi-state system with simple structure, show that the technique of contractors helps to find an imprecise availability. The obtained imprecise availability is more conservative than the BUGF and IUGF methods when we compare the obtained imprecise availabilities with the one we found it by applying the exact method. However, results by using the technique of contractors and the one obtained by BUGF are very close, but the main benefit in our proposed methodology is that we don not have to transform the repair rates and the failure rates into probabilities first and it do not cost a lot of time like the case of UGF.

To study the case when the system become more complex and larger, we took a system with a large number of states in the second application. To simplify the problem, we considered that we have independent components. With independent components, we can figure out the availability of each component then formulate the system's availability.
By comparing the results obtained from the technique of contractors to those obtained from the exact method and the precise method, we found that the imprecise availability ($1-$ unavailability) by the technique of contractors contains the precise availability and the imprecise availability obtained by the exact method. That gives the conclusion, that the technique of contractors works even when the system became larger and complex.

Note that the number of contractions that we should make to achieve the convergent Π, increases when the number of equations in eq. 4.1 increases. Which means when the the transition matrix Q and the steady probability vector Π become larger. In other words, when the system become larger and more complex, depending on the number of components and their states. However, the cost of time isn't excessive compared to other simulation methods (Monte-Carlo simulation). We can reduce the computing time if we want, when the components of the system are independent, in this way we can model for each component its Markov chain and estimate each availability. Finally, we can compute the availability of the system from the availabilities of the components.

Usually, the failure rates are lower than the repair rates. However, in all the different examples with all the possible failure rates and repair rates that we took in our study, we noticed that the ratio $\frac{\lambda}{\mu}$ must always be different to one, so that the technique of contractors works.

Since we are dealing with Markovian approach, we should take into account that when the number of states increases in a big way, we will have the problem of combinatorial explosion, that is why in this case we will make some approximations to simplify the problem.

Markov models, for our best of knowledge, are the most suitable methods, especially when the given data are imprecise. The use of interval Contraction method (Forward-backward contractor) to calculate the availability is very helpful in an easy and guaranteed way.

By using this method, we can extend to more complicated cases and try to find the best configuration in terms of cost and availability. In the next chapter, we aim to do that. We will propose a methodology that helps us to find the best system between many systems, that have the best availability with the best cost.

Chapter 5

Safe design of MSSs

5.1 Introduction

In the previous chapter, we introduced a new computing methodology for availability assessment of an MSS in the presence of uncertainties. These uncertainties are modeled in term of intervals..

To find the availability of the system, Markov model was used. We constructed the Markov model of the system and found out its transition matrix Q. Then, we solved the system of equation

$$\Pi.Q = 0$$

to find Π the vector of interval probabilities of the system's states

$$\Pi = [\pi_1, \ldots, \pi_m]$$

with m is the number of the states of the system.

To find Π, we used the "technique of contractors", which is well-known in the domain of interval analysis. The main idea of this technique is to contract the intervals of probabilities π_is of the system to be in a given state i, into smaller intervals.

When finding the vector Π, we can calculate the availability by doing the sum over all the intervals π_j, where j presents the states where the system works.

We should mention that if the components of the system are independent, we can model each component as a Markov model and we find its availability in order to find the availabil-

ity of the entire system using the connection between the components.

In this chapter, we propose a general methodology to solve the problem of optimal design of MSSs. First, we present the optimization problem and some of the related works that have been done in dependability problem. There are several ways to study the optimal design. However in this thesis, we present two concepts to find the optimal design.

The first one, consists in finding the best architecture among n configurations formed with a fixed number of components k, in the aim to have a better availability with the best cost. The overall cost of the MSS is thus optimized. As soon as we get the optimal availability and optimal cost, we can build the different optimal architectures corresponding to the availability and the cost obtained.

The second one, consists of searching for the same purpose, however in this method we fix the desired architecture of the system formed of k components. Suppose that we have N components, each time we choose k components from the N that forms an architecture. We find the architecture that gives the best availability with the best cost of components, and this architecture will be the optimal one among the others.

To understand the two concepts, we present two numerical examples that will explain the two concepts. To end up with the chapter's conclusion.

5.2 Optimization in dependability

The competition in the industrial world and the user requirements has increased, which made the configuration of the systems becomes more complex. Therefore, reducing the cost with reliable systems is a great challenge [60, 40].

As we described before, reliability and availability are both two concepts in RAMS for system performance. Reliability is the probability that the system will perform for a given period of time, and availability measures the online level of the system [40, 127].

To improve the performance of the system, we can apply different options such as: increasing the reliability or the availability of the components, using active/parallel redundancy, applying preventive/corrective maintenance, using large safety factors, managing dependent failures with the other components, and integrating a monitoring process or di-

agnosis process [60]. However, we need to find a balance between the system cost and the required performance, since we are searching for reliable system without being expensive.

In this context, several studies ([59], [127], [73], [126],. . .) has been made in the aim to optimize the reliability or the availability of the system taking into account the cost of the system.

The optimization of reliability or availability is an interesting goal to researchers and industrialists since the 1960s. In fact, taking into account the reliability or the availability of components during system design makes it possible to to have an accurate idea of the cost and the total reliability or the availability of the system [97].

Kuo et *al.* [60] classify the methods of reliability optimization according to:

- The structure of the system: series, parallel, series-parallel, parallel-series, k out of n systems, undefined systems,. . .
- The type of problem that we are dealing with.
- The techniques of optimization: heuristics, meta-heuristics (genetic algorithm), exact algorithm (dynamic programming, implicit enumeration,. . .) and other methods.

For the part of optimization, researchers have developed and have improved many methods and algorithms. These methods can be divided into two main categories: the exact methods that guarantee the achievement of an optimal solution for problems with reasonable size, and the approached methods (Heuristics and meta-heuristics) that provide good solutions, without any guarantee of optimality, but with shorter calculation time. The exact methods are based on enumeration, all the solutions are in the search space. The approached methods use rather random processes in the exploration of potential solutions, and this in the purpose of dealing with the combinatorial explosion generated when using the exact methods. However, in our study we will not use these approaches.

For optimization, we need to model the system that we have and present the method that will help us to calculate the availability or the reliability. To design a system, we start from a set of components to choose the best among them, in terms of performance and cost. When having the components we formulate the system. That is why before seeking to optimize, it is necessary to apply a method for reliability or availability assessment (in our work, we are interested about availability) and we are using the Markovian approach.

In these works [123, 42, 19], they presented an optimization method of the the parameters of dependability using Markovian model. To our best of knowledge, there is no works that present an optimization method of the imprecise availability using Markovian model. However, for optimization of the system's dependability parameters (availability, reliability,. . .) with uncertainties using different methods exists in the literature [67, 8, 25], but the use of Markovian models is the distinctive aspect of this thesis.

5.3 Two concepts to find the optimal system

When we want to buy a product from the commercial market, we are interested in the product that gives us the best life time. Also, the price of the product is important to us, we do not want to have a product for a big amount of money.

This principle is applied in all domains. This case is similar if we have a system of components. We want to choose the best components in terms of availability and price, to form the best system's configuration. However, in our study we introduced the presence of uncertainties of the failure rates λ and the repair rates μ of the components. To model these uncertainties, we use interval analysis and make all the data concerning the components and the system in term of intervals.

To model the system that we have, we use the imprecise Markovian approach. In consequence, by solving the system of equations we obtain the imprecise availability of the system in terms of interval. The system's availability which is in form of interval, is computed by using the methodology presented in the previous chapter.

The idea of optimization in our work, is to find the best safe design of the system. The best system is when we can balance between the cost and the availability of the system. To achieve this aim, we need to find the best interval of availability with the minimal cost. The best design of the system, is when having the following two criteria:

- The best interval availability of the system $A_s = [\underline{A}_s, \overline{A}_s]$: we need to maximize the lower bound of the interval $\underline{A}_s$ and minimize the length of the interval $(\overline{A}_s - \underline{A}_s)$.
- The best system's cost C_s: we need to minimize the system's overall cost, which is the sum of components cost.

The powerful side of the system's availability interval, is its lower bound $\underline{A}_s$ because our main benefit is to maximize the availability as much as we can. Therefore, we need to

maximize the lower bound $\underline{A}_s$. However, we are interested in finding the best availability interval with the minimal imprecision. Which can be done by reducing the length of the interval A_s, that is why we need to minimize the length of the interval. The total cost of the system C_s is equal to the sum of the costs of its components, so it is obvious that we want to minimize the cost of the system.

To find the best system with the corresponding components, we define our objective function in the aim of its maximization. The objective function is a function that minimizes the cost of the system since we tend to find a lower price of components. The function minimizes the length of the availability interval of the system, to reduce the interval. Finally, it maximizes the lower bound of the availability interval, because we are searching for the best system with the best availability. Our objective function will be noted as *f*.

In this section, we present two points of view or two possible cases to find the optimal system. To find the best safe design of a system, there are a lot of points of view. Each point of view represents a way of study, however in this thesis we chose two of them. The chosen two points of view are widely treated cases and represent the most common case in the real life. Also, these two points of view are two simple cases of optimization before passing to another high level of optimization with more complicated cases. The basic idea of the first case, is to choose the best architecture among m available architecture or systems of a certain number k of components, where this architecture gives the best availability with the minimal cost. The idea of the second case, is to choose the best n components from the N available components to give the best availability of the system with a suitable system cost.

To study the two cases, first we need to define all the information concerning the system that we work on. Suppose that's the system with a specific configuration, is formed from n components. We present all the N available components with their intervals of failure rates λ, their intervals of repair rates μ and their costs. We need to present the different states for each component. We define the configuration of the system (parallel, series, series-parallel,. . .) that we have.

The components of the system could be independent. When this is the case, we find for each component its Markov chain, then its transition matrix, and using the contraction technique we find its interval availability. In the second case the architecture of the system is fixed, we can form the expression of the availability of the system in term of the availabilities of the components. For the cost of the system, simply we sum over the costs of the n

components. However, when the components are dependent, we find the Markov chain of the whole system, then its transition matrix and the availability of the system are assessed based on the methodology of the contraction technique. The same thing is done for the first case of study, however here we are obliged to use the block diagram method with Markov approach, to model the architectures of the systems.

5.3.1 The first point of view

One of the undeniable steps in the safe design of systems is the problem of choosing the best system configuration in the most effective way so as to maximize the overall system's availability and to minimize the overall system's cost. The main objective in this part is to propose a methodology of optimization of the availability of MSSs with multi-states components in presence of both aleatory and uncertainties.

The problem is formulated as follows: let us consider a number of configurations m of a system, each configuration consisting of a specific number of components k with different working states. The related data of each component are imprecise,therefore we have imprecise failure rates and imprecise repair rates provided in form of intervals. The objective is to find the best configuration regarding the system availabilities and costs.

First, we take each configuration of k components and we compute its imprecise availability by using the method introduced in chapter 4, based on Markovian approaches combined with interval contraction techniques. Each configuration of the system will have a different structure (parallel, series, series-parallel, complex,...), with different choices of component characteristics (imprecise failure and repair rates). Then, we compute the overall cost of each configuration. When having the availability and the cost of all the configurations, we define an objective function in terms of cost, lower and upper bounds of availability, and imprecision (length of availability interval). Then, this function is computed for each configuration (with its availability and cost). According to different criteria as high availability, low cost, or low availability imprecision, we will find the best system configuration when we find the maximum objective function *f*. Figure 5.1 illustrates our proposed methodology.

Here, the different steps of our methodology for optimal design of MSS among many configurations:

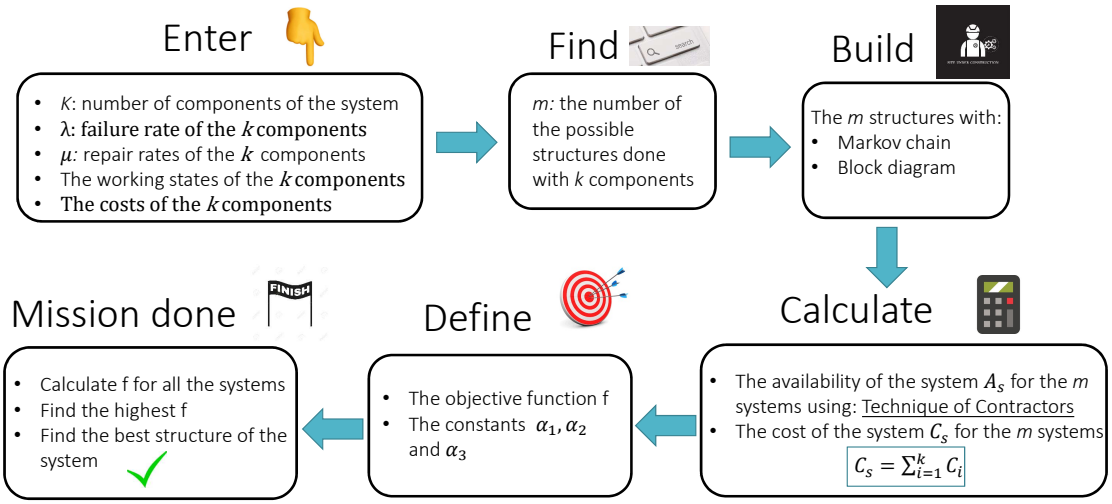


Figure 5.1: Methodology of the first point of view of safe design

1. Define the different system structures that we have:
   - The number of components in each structure.
   - Failure rates and repair rates.
   - The working states of each component for each structure.
   - The different costs for each component in each structure.
2. Our aim is to choose the components and their connection that will achieve the optimal MSS structure with minimal overall cost. Thus, we need to know the number of the system configuration m.
3. Modeling the m structures of each MSS by a Markov chain (or a block diagram, just to understand what type of system it is).
4. Computing the m interval of the system availability A_s for each configuration. We use the method defined in chapter 4. If the components are independent, we can find the availability of each component then we formulate the availability of the entire system regarding the connections between the components. If it is not the case, we calculate the availability of the system by modeling the Markov chain of total system and finding the transition matrix of the system.
5. Computing the m cost C_s.

$$C_s = \sum_{i=1}^{k} C_i \tag{5.1}$$

with k is the number of components for a structure and C_i is the cost of the ith component.

6. The best interval availability of the best system must have the highest value of its lower bound in condition of having the smallest length.

7. Calculate the objective function *f* of each configuration, and find the maximum value which correspond to the best one.

Our aim is to find the best system with the components that gives the best availability with a low cost. Therefore, we need to optimize the cost and the availability interval. We optimize the availability interval of a system, by maximizing its lower bound and minimizing the length of the availability interval. Thus, we need to minimize the cost of the system.

The objective function is in terms of the sum of the total cost of the components, the average of the availability and the lower bound of the availability interval, since it is important to define the best availability with the lower cost. The objective function is given by:

$$f = -\alpha_1 \boldsymbol{C}_s/\beta - \alpha_2(\overline{A}_s - \underline{A}_s) + \alpha_3 \underline{A}_s \tag{5.2}$$

With α_1, α_2 and α_3 are constants, such that $\sum_{j=1}^{3} \alpha_j = 1$.

These constants are defined by the user depending on his point of interest, is he want to focus on the cost or on the availability.

The cost is by a price unit, the factor β is a number applied on the cost to make the three factors (cost, availability, length of the interval availability) in the same scale.

5.3.2 The second point of view

In the previous subsection, we explained a method of how to choose the best system in terms of high availability and minimal cost with the presence of uncertainties. We searched for the best system between m possible systems formed from k components. We defined the objective function *f*, which will be calculated for the m structures so that we can choose the best system corresponding to the highest value of *f*.

That is one of the possible points of view of the cases that we can deal with in real life to find the best system. In this section, another case of study that we can have it will be presented.

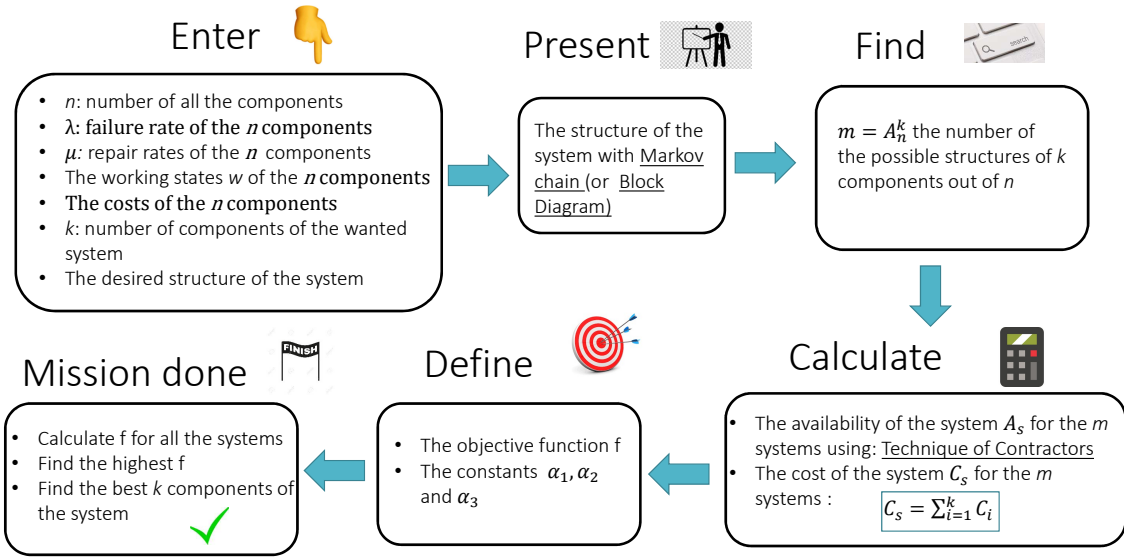


Figure 5.2: Methodology of the second point of view of safe design

Suppose that the producer of some system has the idea of the structure of the system of the product that he want to build. However, the producer do not recognize what are the best components to be used in the constructing of the system.

The difference between the previous case and this case, is that here we have type of the structure of the system by using the block diagram. The problem is to find the best architecture of k components. In this case, we search for the best system among m configurations. Each configuration is formed from k components chosen from a number n components that the producer could have. In other words, we need to find the k components from the available n components in the market that lead us to the best system.

We should always remember that the best system or the optimal system, is the system where we have a better availability with the best cost and minimal availability imprecision.

In this aim, we seek to minimize the overall cost of the MSS, and maximize the availability. Maximization of the availability interval is done when we minimize the availability interval imprecision and maximize the lower bound of the interval, as we explained in the first point of view. As soon as we get the optimal availability interval with the optimal cost, we can find the optimal architecture corresponding to the available and the cost obtained.

Here, we present the different steps of our methodology for optimal design of MSS for this case of study (cf. Figure 5.2):

1. Introduce all the information that we have concerning the system and the available components:

   - The number of all the possible components that we have, denoted by n. We choose a certain number of them to form the system.
   - Failure rates λ and repair rates μ for each component of the n components.
   - The number of the possible states of each component of the n components (binary component or multi-states components).
   - The working states w of each component of the n components.
   - The different costs for each component of the n components.
   - The number of components k to build the structure of the system to be optimized.
   - The type of structure of the system chosen (parallel, series-parallel,. . .).

2. Our aim is to choose the best k components that will achieve the optimal MSS structure with minimal overall cost C_s and best availability interval. We mean by the best availability interval of the system, when having the best availability interval in terms of maximal lower bound and minimal imprecision.

3. Modeling the structure of the MSS by a Markov chain (or a block diagram, just to understand what type of system it is).

4. If the components are supposed to be dependent, at each time we take k components out of the n components and we formulate the Markov chain of the system. We form the transition matrix Q. By applying the method introduced in Chapter 4, we solve the system of equation $\Pi.Q = 0$, so that we get the availability of the system by doing the sum over all the π_j, where j presents the working states of the system.

5. If the components are independent, we can calculate the availability of each component apart. We use the method defined in Chapter 4. To compute the availabilities for the n components, we form the transition matrix of the ith component then we solve the system of equation of the component i $\Pi.Q = 0$. To obtain the availability of each component i, we do the sum over all the π_j the probabilities of the working states of the component i.

6. Since the components are independent, we find the relation between the availabilities of the k components to form the availability A_s of the MSS

$$A_s = f(A_1, \ldots, A_k) \tag{5.3}$$

A_l with $l = 1, \ldots, k$ is the availability of the component l.

7. We define all the possible systems for the structure that we defined. For each system p, we take a number k out of the n components, with each component is different from the other one. We will have m architectures (of systems), with m is equal to the arrangements A_n^k.

8. In the case of independent components, for each system p, we compute its availability of the system A_{s_p} by applying the relation 5.3 between the k components and the overall system. For dependent components, simply for each system p, we find the transition matrix of the whole system and then solve the equation $\Pi.Q = 0$. In order to find the availability of the system, we sum over all the probabilities the functional states.

9. However the components are dependent or not, we need to calculate the cost of the total system C_{s_p}. We calculate the cost of each system p by doing the sum of the costs of the k components

$$C_{s_p} = \sum_{l=1}^{k} C_l \tag{5.4}$$

with $p = 1, \ldots, m$.

To ensure that the constraints imposed are respected, we should ensure that the real exact availability A_{real} of the MSS, belongs to the availability interval of the best system.

For that reason, we should add a constraint representing the availability imprecision (the length of the interval which is the difference between the lower bound and the upper bound of the system availability interval).

The parameter α_1 makes it possible to minimize the cost of the system.
The parameter α_2 to minimize the imprecision of the availability interval of the system.
The parameter α_3 to maximize the lower bound of the availability interval of the system.

Our aim is to find the best system with the components that gives the best availability with a low cost and best availability interval (depending on the lower bound of the interval and its imprecision). Therefore, we need to reduce the cost of the system and reduce

the availability interval. We maximize the system's availability interval, by maximizing its lower bound and reducing the length of the system's availability interval.

Same as the first case, the objective function f is in terms of the sum of the total cost of the components, the imprecision of the system's availability and its lower bound, since it is important to define the best availability with the lower cost.

The objective function f is defined by:

$$f = -\alpha_1 \boldsymbol{C}_s/\beta - \alpha_2(\overline{A}_s - \underline{A}_s) + \alpha_3 \underline{A}_s \tag{5.5}$$

With α_1, α_2 and α_3 are constants, such that $\sum_{j=1}^{3} \alpha_j = 1$.

We choose these coefficients according to what the user asks for. For example, if he wants to minimize the cost more, we choose a high value for α_1.

The cost is by a price unit, the factor β is a number applied on the cost to make the three factors (cost, availability, length of the interval availability) in the same scale.

5.4 Numerical examples

In the previous section, we presented two points of view of optimization of the imprecise availability of an MSS. We formulated our two methodologies of choosing the best system among m possible systems taking into account the best system's availability interval in terms of the best value of the lower bound of the interval and the best length for the interval, and of course the best cost of the system.

To illustrate our two proposed methodologies, in this section, we present two numerical examples. The first example is applied using the first point of view and the second example is by applying the second point of view. The two numerical examples represent two cases to illustrate the two optimization methods, however the two methodologies are well applied on different cases and types of systems.

5.4.1 Example of the first point of view

As we described the point of view and the methodology in 5.3.1 and to a better explanation, here a simple example that presents the case.

Component i	**Failure rate** $\times 10^{-3}$ *per hour* (λ_i)	**Repair rate** $\times 10^{-2}$ *per hour* (μ_i)	**Cost** *per Euros* (C_i)
1	[3,4]	[2.5, 3.1]	1500
2	[4, 4.8]	[2, 3]	1200
3	[3.5, 4.1]	[3, 3.5]	1700
4	[2, 2.8]	[2, 2.8]	2000

Table 5.1: Given failure rate, repair rate and cost for each component

The main idea of this study, is to find the best system among m possible systems, all of them are formed of k components. Each component has different degraded states from perfect functioning to total failure, with its specific imprecise failure rates λ and imprecise repair rates μ. Each configuration of the system will have a different structure (parallel, series, series-parallel, complex,. . .).

Suppose that the producer is searching for the best system of 4 components. To simplify we will consider that the 4 components are binary, where all the information: failure rates λ per *hour*, repair rates μ per *hour* and the cost *C* per *Euros*, for each component are represented in Table 5.1.

From the four components, we can build 40 different architectures of the system. We eliminate the repeated architectures, for example the parallel system of components 1, 2 , 3 and 4 is the same parallel system of components 3, 1 ,4 and 2, and so on.

To find the best system from the 40 possible architectures, we need to define the objective function *f*. We want to maximize the availability, and that is done by maximizing the lower bound and minimizing the length of the interval of the availability, and we need to minimize the cost. To make all the three factors at the same scale, we will take the value of β equal to 10^4.

The corresponding objective function is:

$$f = -\alpha_1 \boldsymbol{C}_s/10^4 - \alpha_2(\overline{A}_s - \underline{A}_s) + \alpha_3 \underline{A}_s \tag{5.6}$$

The cost of each system is obtained by doing the sum of the costs of the 4 components. In this example, we have the same 4 components for the 40 systems so the cost of the system $\boldsymbol{C}_s$ is:

$$\boldsymbol{C}_s = C_1 + C_2 + C_3 + C_4 = 1500 + 1200 + 1700 + 2000 = 6400 \text{ Euros}$$

Component i	**Availability** A_i
1	[0.8532, 0.9174]
2	[0.7851, 0.8953]
3	[0.8752, 0.9124]
4	[0.8686, 0.9379]

Table 5.2: Availability of the 4 components

In this example, we consider that the components are independent. Therefore, we calculate the corresponding availability of each component apart using the method of contraction introduced in Chapter 4. Table 5.2 presents the availabilities of the components.

The 40 possible architectures of the system of 4 components with simple structure, are presented in Figure 5.3. Structures 1 and 2 have one case each. Structures 3 and 7 has each 4 possible cases by taking all the possibilities of occupation of a, b, c and d by the 4 components with the elimination of the repeated architectures. Structures 4 and 9 have each 6 possible systems. Structures 5 and 6 have each 3 possible systems. To end with the structure 8, with 12 possible systems.

Each system of the 40 systems will have different interval availability A_s. It is calculated depending on the relation between the components in the concerning structure of the system.

To calculate the objective function, we use eq. 5.6. We choose the values of α_1, α_2 and α_3 according to what criterion we are interested in more than the others.

As example, let us suppose that we are focusing more on maximizing the lower bound of the interval of the system's availability. We take $\alpha_1 = \alpha_2 = 0.1$ and $\alpha_3 = 0.8$. Table 5.3, shows all the availabilities of the systems with the corresponding values of the objective function.

From Table 5.3, we find that the best result is:

$$\begin{cases} \text{The best objective function: } f = 0.7355 \\ \text{The best availability: } A_s = [0.9994, 0.9999] \end{cases}$$

That means that the system with the **parallel** structure is the best architecture. We notice that even when we change the values of the parameters α_i, the parallel system has always the

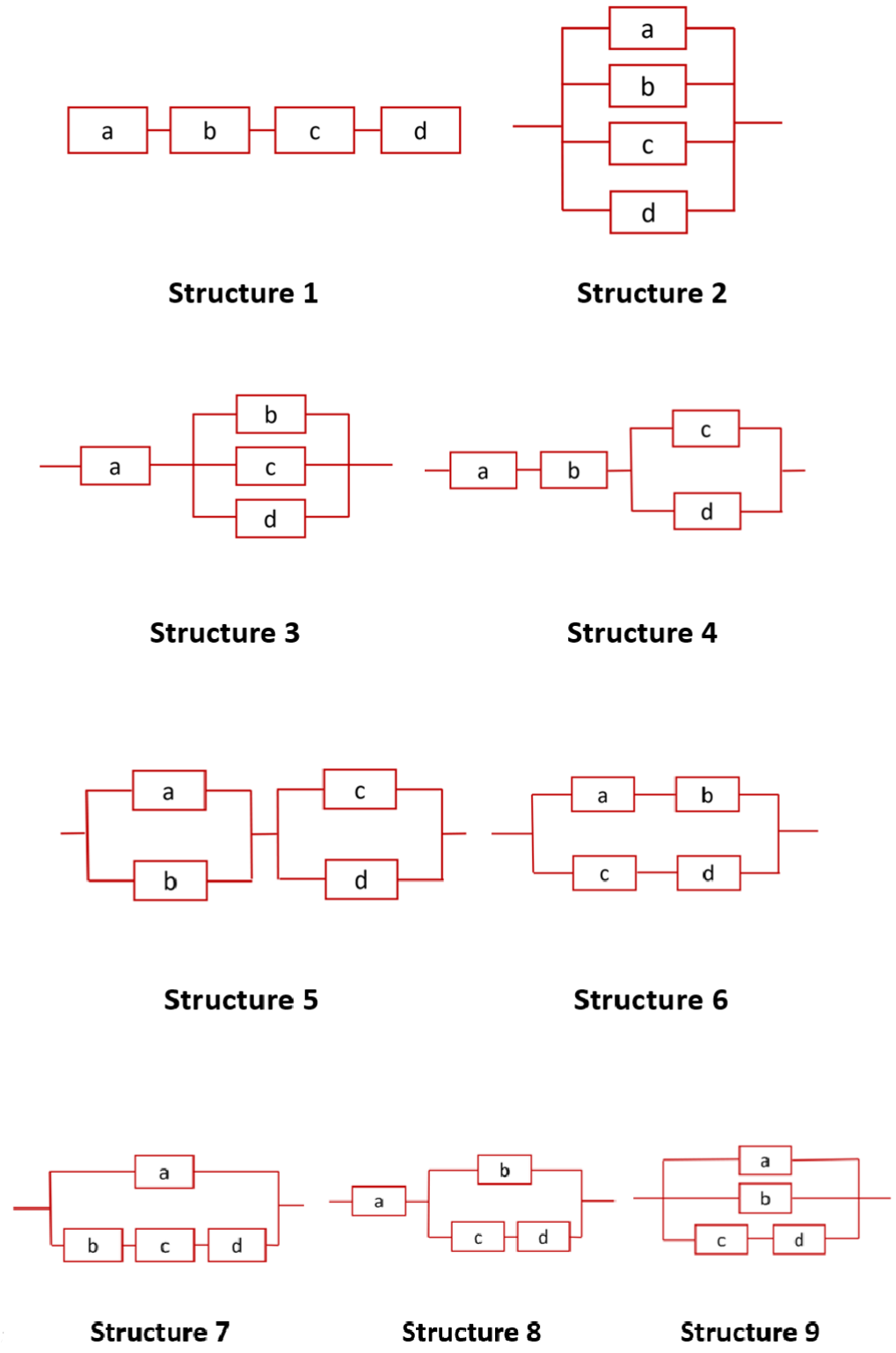


Figure 5.3: The 40 possible cases of system's structure with the 4 components

System j	**Availability** A_{s_j}	**Objective function** f_j
1	[0.5093, 0.7029]	0.3241
2	[0.9994, 0.9999]	**0.7355**
3(1): a=1; b=2; c=3; d=4	[0.8502, 0.9169]	0.6094
3(2): a=2; b=1; c=3; d=4	[0.7832, 0.8949]	0.5514
3(3): a=3; b=1; c=2; d=4	[0.8716, 0.9119]	0.6293
3(4): a=4; b=1; c=2; d=4	[0.8652, 0.9372]	0.62101
4(1): a=1; b=2; c=3; d=4	[0.6589, 0.81693]	0.44732
4(2): a=1; b=3; c=2; d=4	[0.7257 ,0.8316]	0.5059
4(3): a=1; b=4; c=2; d=3	[0.7213, 0.8526]	0.4999
4(4): a=2; b=3; c=1; d=4	[0.6739, 0.8127]	0.4612
4(5): a=2; b=4; c=1; d=3	[0.6695, 0.8336]	0.4552
4(6): a=3; b=4; c=1; d=2	[0.7363, 0.8484]	0.51389
5(1): a=1; b=2; c=3; d=4	[0.9525, 0.9859]	0.6947
5(2): a=1; b=3; c=2; d=4	[0.9539, 0.9863]	0.6959
5(3): a=1; b=4; c=2; d=3	[0.9544, 0.9857]	0.6964
6(1): a=1; b=2; c=3; d=4	[0.9298, 0.9742]	0.6673
6(2): a=1; b=3; c=2; d=4	[0.9194, 0.9739]	0.6615
6(3): a=1; b=4; c=2; d=3	[0.91904, 0.9744]	0.6656
7(1): a=1; b=2; c=3; d=4	[0.9498, 0.9807]	0.6846
7(2): a=2; b=1; c=3; d=4	[0.9245,0.9775]	0.6703
7(3): a=3; b=1; c=2; d=4	[0.9478, 0.9799]	0.69108
7(4): a=4; b=1; c=2; d=3	[0.9456, 0.9844]	0.6886
8(1): a=1; b=2; c=3; d=4	[0.8092, 0.9035]	0.5739
8(2): a=1; b=3; c=2; d=4	[0.8193, 0.9045]	0.5829
8(3): a=1; b=4; c=2; d=3	[0.8181, 0.90701]	0.5816
8(4): a=2; b=1; c=3; d=4	[0.7575, 0,8846]	0.5292
8(5): a=2; b=3; c=1; d=4	[0.7597, 0.8843]	0.5313
8(6): a=2; b=4; c=1; d=3	[0.759,0.8862]	0.5304
8(7): a=3; b=1; c=2; d=4	[0.8344, 0.9003]	0.5969
8(8): a=3; b=2; c=1; d=4	[0.8266, 0.8991]	0.59003
8(9): a=3; b=4; c=1; d=3	[0.8373, 0.9023]	0.5993
8(10): a=4; b=1; c=2; d=3	[0.8288, 0.9237]	0.5895
8(11): a=4; b=2; c=1; d=3	[0.8214, 0.9219]	0.58308
8(12): a=4; b=3; c=1; d=2	[0.8329, 0.9232]	0.5933
9(1): a=1; b=2; c=3; d=4	[0.9924, 0.9987]	0.7293
9(2): a=1; b=3; c=2; d=4	[0.9941, 0.9988]	0.7308
9(3: a=1; b=4; c=2; d=3)	[0.9939, 0.99906]	0.7306
9(4): a=2; b=3; c=1; d=4	[0.99306, 0.9987]	0.7298
9(5): a=2; b=4; c=1; d=3	[0.9928, 0.9989]	0.7296
9(6): a=3; b=4; c=1; d=2	[0.9945, 0.99903]	0.7312

Table 5.3: Availability and objective function of the 40 systems in Figure 5.3

Component i	**Failure rate** $\times 10^{-3}$ *per hour* (λ_i)	**Repair rate** $\times 10^{-2}$ *per hour* (μ_i)	**Cost** *per Euros* (C_i)
1	[3,4]	[2.5, 3.1]	1500
2	[4, 4.8]	[2, 3]	1200
3	[3.5, 4.1]	[3, 3.5]	1700
4	[2, 2.8]	[1.5, 2]	2000
5	[30,40]	[6, 7]	700
6	[6, 7]	[2, 3]	1000
7	[5, 10]	[2, 4]	1100
8	[7, 9]	[5, 6]	1300

Table 5.4: Given failure rate, repair rate and cost for each component

best objective function, while the rank between the other architectures changes. However, that may change if we take the case of dependent components.

5.4.2 Example of the second point of view

To understand the basic methodology of the second point of view, we give the following simple example.

The main idea of this study, is to find the k components among n possible components that form a demanded type of the system, with the best availability and the best price. Of course, each component has different degraded states from perfect functioning to total failure, with its specific imprecise failure rates λ and imprecise repair rates μ.

Suppose that the producer has 8 possible components. To simplify, we will consider that the 8 components are binary, where all the information: failure rates λ per *hour*, repair rates μ per *hour* and the cost *C* per *Euros*, for each component are represented in Table 5.4.

The producer is searching for best 4 components among the 8 possible components, that form the best system where its structure is represented in Figure 5.4. It is a parallel-series system of four components: a, b, c and d.

From the four components, we have 1680 possibilities of arrangement between the components. That means that we will find the best components that form the best structure among the 1680 cases.

As in the first example, to find the best system from the 1680 possible architectures, we

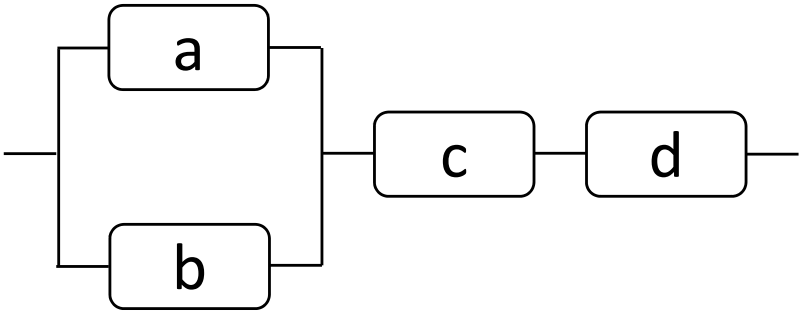


Figure 5.4: The structure of the system of 4 components

need to define the objective function *f*. We want to maximize the availability, and that is done by maximizing the lower bound and minimizing the length of the availability's interval, and we need to minimize the cost. To make all the three factors at the same scale, we take the value of β equal to 10^4.

The corresponding objective function is:

$$f = -\alpha_1 \boldsymbol{C}_s/10^4 - \alpha_2(\overline{A}_s - \underline{A}_s) + \alpha_3 \underline{A}_s \tag{5.7}$$

The cost of each system is obtained by doing the sum of the costs of the 4 components. In this example, and for each system we have different components which means different system's cost $\boldsymbol{C}_s$, with:

$$\boldsymbol{C}_s = C_1 + C_2 + C_3 + C_4 \text{ per } \textit{Euros}$$

In this example, we consider that the components are independent. Therefore, we calculate the corresponding availability of each component apart using the method of contraction introduced in Chapter 4. Table 5.5 presents the availabilities of the components.

Each system of the 1680 systems will have different interval availability A_s. It is calculated depending on the relation between the components, which is:

$$A_s = A_c.A_d\,[1 - (1 - A_a)(1 - A_b)] \tag{5.8}$$

Where a, b, c and d are chosen from the eight components.

To calculate the objective function, we use eq. 5.7. We choose the values of α_1, α_2 and α_3 according to what criterion we are interested in more than the others.

Component i	**Availability** A_i
1	[0.8532, 0.9174]
2	[0.7851, 0.8953]
3	[0.8752, 0.9124]
4	[0.8288, 0.9171]
5	[0.4657, 0.8003]
6	[0.6989, 0.8602]
7	[0.5333, 0.9333]
8	[0.8375, 0.9022]

Table 5.5: Availability of the 8 components

By choosing the values of α_1, α_2 and α_3 depending on the criterion studied, we find the components that represent the best architecture.

In this example, we study the best architecture when we are interested in all the criteria at the same level.
We take the same value for all the parameters with $\alpha_1 = \alpha_2 = \alpha_3 = 0.3333333$.

We find that the highest value of the objective function, is: $f = 0.014964$. With the corresponding availability and cost:

$$\begin{cases} A_s = [0.7102, 0.8256] \\ \boldsymbol{C_s} = C_1 + C_3 + C_6 + C_8 = 5500 \textit{ Euros} \end{cases}$$

It corresponds to the best system, with:

$$\begin{cases} \text{a= component 3} \\ \text{b= component 1} \\ \text{c= component 8} \\ \text{d= component 6} \end{cases}$$

It is the best architecture, in terms of availability interval (in terms of imprecision and lower bound) and cost. If we are looking for a system with same priority for cost, lower bound of the availability interval and its imprecision, we should take this architecture.

Case number	α_1	α_2	α_3	C_s	A_s	f	**System**			
							a	b	c	d
1	1	0	0	4000	[0.3299, 0.7860]	-0.4	7	6	5	2
2	0	1	0	6500	[0.7147, 0.8176]	-0.1029	8	3	4	1
3	0	0	1	6500	[0.7260, 0.8303]	0.72604	3	1	8	4
4	0.3333333	0.3333333	0.3333333	5500	[0.7102, 0.8256]	0.014964	3	1	8	6
5	0.2	0.1	0.7	5700	[0.7207,0.8285]	0.3797	3	1	8	2
6	0.2	0.7	0.1	5700	[0.71001, 0.8161]	-0.1173	8	3	2	1
7	0.7	0.1	0.2	4200	[0.5518, 0.7852]	-0.2069	8	2	6	5

Table 5.6: The best achitectures when changing α_i

Now, let us suppose that the producer is interested more in maximizing the lower bound of the availability. Therefore, the values of the constants are for example $\alpha_1 = 0.2$, $\alpha_2 = 0.1$ and $\alpha_3 = 0.7$. In this case, the best architecture is obtained with:

$$\begin{cases} f = 0.3797 \\ A_s = [0.7207, 0.8285] \\ C_s = 5700 \ \textit{Euros} \end{cases}$$

Such the system is:

$$\begin{cases} \text{a= component 3} \\ \text{b= component 1} \\ \text{c= component 8} \\ \text{d= component 2} \end{cases}$$

Table 5.6 presents some of the possible cases when we change the parameters α_i.

We took this example, and we found out the ranking of the five best structures for each case when we variate the values of α_i. Figure 5.5 presents the obtained rank for this case of study. The objective function is *f*. The relative distance of the other ranks to the obtained structure in the first rank is represented by D, with

$$D = \frac{|(f_{\text{rank } i} - f_{\text{rank 1}})|}{f_{\text{rank } i}} \tag{5.9}$$

Such that $f_{\text{rank } i}$ is the objective function of the structure with rank i and $f_{\text{rank 1}}$ is the objective function of the first place structure.

The structures are represented by $S = \{a, b, c, d\}$, where a, b, c and d take place from the 8 components.

As we can see, for each case we have different ranking. However, there is repeated structures in many cases with different ranks. Here is the most repeated structures.

$$\begin{cases} S = \{3, 1, 8, 2\} \\ S = \{3, 1, 8, 6\} \\ S = \{8, 3, 2, 1\} \\ S = \{3, 1, 8, 4\} \\ S = \{8, 2, 6, 5\} \end{cases} \tag{5.10}$$

Case					
1	f= -0.4 S= {7, 6, 5, 2}	D=0.0243 f=-0.41 S= {8, 7, 6, 5}	D= 0.0476 f=-0.42 S= {8, 6, 5, 2}	D= 0.0697 f= -0.43 S= {8, 7, 5, 2}	D= 0.0909 f= -0.44 S={6, 5, 2, 1}
2	f=-0.1029 S= {8, 3, 4, 1}	D= 0.0128 f= -0.1043 S={3, 1, 8, 4}	D= 0.0303 f= -0.1061 S={8, 3, 2, 1}	D= 0.04507 f= -0.1078 S={3, 1, 8, 2}	D= 0.06397 f= -0.1099 S={8, 3, 4, 2}
3	f= 0.72604 S={3, 1, 8, 4}	D= 0.0073 f= 0.7207 S={3, 1, 8, 2}	D= 0.0093 f= 0.71933 S={3, 1, 4, 2}	D= 0.0158 f= 0.7147 S= {8, 3, 4, 1}	D= 0.0221 f= 0.7102 S={3, 1, 8, 6}
4	f= 0.014964 S={3, 1, 8, 6}	D= 0.0456 f= 0.0143 S={3, 1, 8, 2}	D= 0.1178 f= 0.0133 S= {8, 3, 6, 2}	D= 0.1917 f= 0.01255 S={8, 3, 6, 1}	D= 0.3266 f= 0.0112 S={8, 3, 2, 1}
5	f= 0.3797 S={3, 1, 8, 2}	D= 0.0108 f= 0.3756 S={3, 1, 8, 6}	D= 0.0197 f= 0.3723 S={8, 3, 2, 1}	D=0.0285 f= 0.3692 S={8, 3, 6, 1}	D= 0.031 0.3683 S={3, 1, 6, 2}
6	f= -0.1173 S={8, 3, 2, 1}	D= 0.00064 f= -0.11739 S={3, 1, 8, 2}	D= 0.0147 f= -0.119 S={8, 3, 6, 1}	D=0.0202 f= -0.1197 S={3, 1, 8, 6}	D= 0.0485 f=-0.1233 S= {8, 3, 6, 2}
7	f= -0.2069 S={8, 2, 6, 5}	D=0.0352 f= -0.2145 S= {8, 6, 5, 2}	D= 0.0398 f= -0.2155 S={8, 1, 6, 5}	D= 0.0556 f= -0.2191 S={6, 2, 8, 5}	D=0.0557 f=-0.2192 S={2, 1, 6, 5}

Figure 5.5: Ranking of the structures for the cases in Table 5.6

	$S = \{3, 1, 8, 2\}$	$S = \{8, 3, 2, 1\}$	$S = \{3, 1, 8, 6\}$	$S = \{8, 3, 4, 1\}$	$S = \{3, 1, 8, 4\}$
$D_{average}$	0.0882	0.1319	0.1548	0.4406	0.494
Rank	1	2	3	4	5

Table 5.7: The average of the distance between one structure and the best structure

The cases in Table 5.6, presents a general view of all the possible cases that we could have. When we take the extreme cases we are focusing on one criterion without the others. The equal parameters α_i presents the case where we are interested in all the criteria as equal. The last three cases, present examples of the other cases that we could have. From Figure 5.5, we found out the most occupied structure in all the cases is the system $S = \{3, 1, 8, 2\}$. However, to see how much this structure is close to the first rank, we calculated the average of the distance of the corresponding structure to the structure of the first in each case. Table 5.7 presents the average of the distance $D_{average}$ between one structure and the best structure for the 5 systems in 5.10. Note that we cannot compare the results obtained by the two points of view since each one has different hypothesis.

As we can see, from Table 5.7 that the best system is for $S = \{3, 1, 8, 2\}$. This structure for any case of study, can be chosen since it always classified in the the list of the best structures.

5.5 Conclusion

In the previous chapter we introduced a method to calculate the imprecise availability of a MSS with multi-states components. This method is applied by using Markovian model with the use of a technique applied in Interval Analysis called "Technique of contractors". In this chapter, our aim is to use this proposed method to find the best design of a system.

The best design is chosen by taking into account the minimal cost of the system C_s, which is the sum of the costs of all the components that form the system, and the system's availability interval $[A_s]$. Which means to choose the best design, we must minimize the cost of the system and maximize the availability of the system. The latter is done by minimizing the imprecision of the interval of A_s, which is the length of the interval and maximizing the lower bound of the interval.

For these arguments, we defined an objective function *f* to be maximized, see eq. 5.7. The best structure is chosen for the highest value of *f*. Where the parameters α_1, α_2 and α_3 are constants chosen depending on which criterion the producer is focusing on: the cost, the

imprecision of $[A_s]$ or the the lower bound of $[A_s]$.

In this chapter we presented two possible points of view:

- The first one: Finding the best structure of k components, among m possible structures.
- The second one: Finding the best k components among n components that give the best structure, where the type of structure is defined before.

With two simple examples, we detailed the methods of the two points of view. To simplify we took binary components, however we do the same procedures when we have multi-states components and when having much complex systems. We studied the change of the ranking when we take different cases of α_i.

To verify the methods, we can study the different cases when we fix one parameter (by fixing α_i) and modify the others.

With these two methods we can always find the best structure, with the minimum cost and the maximum availability. In addition, it is a general description for the three criteria. These two methods can be applied when having multi-states components and when having large and complex systems with a big number of components.

Conclusions and perspectives

The research work presented in this thesis, deals with the problem of availability assessment of a multi-states systems (MSS) taking into account the uncertainty of the components data (failure rates and repair rates), when time tends towards infinity. Another problem presented in this thesis, which is how to find the best design of a system among different structure and which components must take into account.

1. Conclusion

Throughout this thesis, we have tried to find a methodology which will optimize the availability of MSSs. However, there are few methods in the litterateurs that deal with the case of such systems. In particular, traditional works was interested in the availability of the binary systems (normal working and total failure). Nevertheless, for a wide range of real systems the binary approach no longer makes sense because in real life the systems will have many possible states.

Also in real life, the availability of the systems is not precise because uncertainties on components data will occur. These uncertainties could be about the components performance values, they may be imprecise. Or, uncertainties could be about the failure rates λ and the repair rates μ which may be imprecise because the estimation of the transition rates is difficult (new elements, rarely affected elements, expensive elements,. . .) or they may not be constant in time,. . . In other works, they applied different methods to model the uncertainties and to calculate the availability or the reliability of the system (IUGF, BUGF,Monte Carlo simulation,. . .). However, the methods in the presented works are limited in their use. These methods can be applied for particular systems (simple systems), small systems (with a small number of components and states), also their computing time is large.

In this context, this thesis allowed us to identify an original methodology for the imprecise availability assessment of MSSs. We have proposed the use of imprecise Markovian

approaches in presence of imprecise failure and repair rates. Markovian approaches are suitable methods to present the systems states and to calculate their steady probabilities. However, the use of Markovian methods is limited since we will have a combinatorial explosion when the number of states increases (the number of states increases exponentially with the number of components), in this case some approximation can be applied to avoid such problem. The main benefit of the Markovian methods is that we can calculate the availability when t tends to infinity. The user is always looking for: "What is the probability that the system remains functional after a long time?".

In this thesis, we chose to model the failure rates, the repair rates and the systems states initial probabilities in terms of intervals. Therefore, the transition matrix will be an interval transition matrix. To compute the imprecise availability, we have to know the systems states probabilities. These probabilities are obtained when solving the system of equations:

$$\Pi.Q = 0$$

In this aim, we proposed to use a technique in interval analysis called the technique of contractors. In particular, the forward-backward contraction method on the equations obtained by the transition matrix of the MSSs. The main goal of the contraction method is to reduce the initial intervals of the systems states probabilities to their most possible minimum sizes. We tested the method on several examples, in this thesis we mentioned two of them. For the first example, we compared the results of the imprecise availability obtained by the methodology to the ones obtained by the exact method and the IUGF and BUGF. For the second example, our aim was to increase the number of components and the number of states and compare the results of the methodology to the results obtained by the exact method and the precise one. We found that the interval obtained by the technique of contraction is more conservative than the interval obtained by the two methods IUGF and BUGF, however it is very close to the one obtained by the BUGF. We found that the precise availability is always belong to the intervals of the exact method and the technique of contractors. Based on this method, we can study the availability of larger systems, where the number of components and the number of possible states for each element is higher. However, in all of this thesis we supposed that the components are independent, and in this case the technique of contractor do not cost a lot of time in computing, because we find the availability of each component then we figure out the availability of the entire system.

To reach our aim of optimization and finding the best system's design, we proposed two methodologies corresponding to two possible points of view. These two points of view

represent two simple case of study for finding the best system. The first one is when having k a fixed number of system's components, and the problem is to find the best system among all the possible systems of the k components such that we have the best availability and the best cost. The second case, is when having n a number of possible components with a certain type of system of k components, the problem is to find the best k components among the n components such that we obtain the best system's availability with the best cost. In the two cases, we define our objective functions in terms of the system's availability and the system's cost. The best system is characterized by the maximization of the objective function, this is done by maximizing the lower bound of the availability's interval and by minimizing the length of the interval and the system's cost. We presented two numerical example to illustrate the two methodologies, however the two methods are well applied in all kind of systems.

2. Perspectives

All the results obtained in this thesis, are according to some hypotheses: constant failure and repair rates, absence of common causes failure, independent components of the system. In what follows, we will base on these hypotheses by proposing some perspectives for extending the obtained results.

- Applying the proposed methodology using the technique of contractors on systems with dependent components and with the case of common cause failure. In this case we need to model the system with its Markov chain without taking each component apart.
- The use of non-constant transition rates (repair rates and failure rates) with semi-Markov processes (in this case, stochastic Petri nets are used).
- Apply the proposed methodologies on an example in real life.
- Applying the technique of contractors on different methods in dependability, to compare to the results obtained by Markovian methods.
- For the optimization problem, we choose to apply two points of view, however we can expand the problem to more complicated cases. For example: finding the best system of k components among n possible components without fixing a certain type of structure.

Bibliography

[1] Fahed Abdallah, Amadou Gning, and Philippe Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807–815, 2008.

[2] Felipe Aguirre, Sébastien Destercke, Didier Dubois, Mohamed Sallak, and Christelle Jacob. Inclusion–exclusion principle for belief functions. *International Journal of Approximate Reasoning*, 55(8):1708–1727, 2014.

[3] Muhammad Juned Akhtar and Ingrid Bouwer Utne. Human fatigue's effect on the risk of maritime groundings - A Bayesian Network modeling approach. *Safety Science*, 62:427–440, February 2014.

[4] LANNOY André. *Maîtrise des risques et sûreté de fonctionnement: repères historiques et méthodologiques (Collection Sciences du risque et du danger)*. Lavoisier, 2008.

[5] Terje Aven. Interpretations of alternative uncertainty representations in a reliability and risk analysis context. *Reliability Engineering & System Safety*, 96(3):353–360, 2011.

[6] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, et al. *Fundamental concepts of dependability*. University of Newcastle upon Tyne, Computing Science, 2001.

[7] Rolf Backofen. A polynomial time upper bound for the number of contacts in the hp-model on the face-centered-cubic lattice (fcc). *Journal of Discrete Algorithms*, 2(2):161–206, 2004.

[8] André Teófilo Beck and Wellison José de Santana Gomes. A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty. *Probabilistic Engineering Mechanics*, 28:18–29, 2012.

[9] Michael Beer and Edoardo Patelli. Engineering analysis with vague and imprecise information. *Structural Safety*, (52):143, 2015.

[10] Frédéric Benhamou, Frédéric Goualard, Laurent Granvilliers, and Jean-François Puget. Revising hull and box consistency. In *Int. Conf. on Logic Programming*. Citeseer, 1999.

[11] M Bergot and L Grudzien. Sûreté et diagnostic des systèmes industriels. principaux concepts, méthodes, techniques et outils. *Diagnostic et sûreté de fonctionnement*, 5(3):317–344, 1995.

[12] R Billinton and L Wenyuan. Hybrid approach for reliability evaluation of composite generation and transmission systems using monte-carlo simulation and enumeration technique. In *IEE Proceedings C (Generation, Transmission and Distribution)*, volume 138, pages 233–241. IET, 1991.

[13] Roy Billinton and Peng Wang. Teaching distribution system reliability evaluation using monte carlo simulation. *IEEE Transactions on Power Systems*, 14(2):397–403, 1999.

[14] Z. W. Birnbaum, J. D. Esary, and S. C. Saunders. Multi-Component Systems and Structures and Their Reliability. In *Technometrics*, pages 55–77. American Statistical Association, 1961.

[15] David Blockley. Analysing uncertainties: Towards comparing bayesian and interval probabilities'. *Mechanical Systems and Signal Processing*, 37(1-2):30–42, 2013.

[16] Mark A Boyd and Sonie Lau. An introduction to markov modeling: Concepts and uses. 1998.

[17] Alvis Brazma, Inge Jonassen, Ingvar Eidhammer, and David Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of computational biology*, 5(2):279–305, 1998.

[18] Patrick O Brown and David Botstein. Exploring the new world of the genome with dna microarrays. *Nature genetics*, 21(1s):33, 1999.

[19] Gustavo Callou, Joao Ferreira, Paulo Maciel, Dietmar Tutsch, and Rafael Souza. An integrated modeling approach to evaluate and optimize data center sustainability, dependability and cost. *Energies*, 7(1):238–277, 2014.

[20] Gilles Chabert. A very short introduction to interval analysis. *Seminaires doctorants 2*, page 5, 2006.

[21] Pat L Clemens. Fault tree analysis. *JE Jacobs Severdurup*, page 13, 2002.

[22] Commission of the European Communities. Information Technology Security Evaluation Criteria, Harmonized criteria of France, Germany, the Netherlands, the United Kingdom. Technical report, 1991.

[23] Hend Dawood. *Theories of interval arithmetic: Mathematical foundations and applications*. LAP Lambert Academic Publishing, 2011.

[24] Gert De Cooman, Filip Hermans, and Erik Quaeghebeur. Imprecise markov chains and their limit behavior. *Probability in the Engineering and Informational Sciences*, 23(4):597–635, 2009.

[25] Kalyanmoy Deb, Shubham Gupta, David Daum, Jürgen Branke, Abhishek Kumar Mall, and Dhanesh Padmanabhan. Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 13(5):1054–1074, 2009.

[26] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 57–72. Springer, 2008.

[27] Armen Der Kiureghian and Jyh-Bin Ke. The stochastic finite element method in structural reliability. *Probabilistic Engineering Mechanics*, 3(2):83–91, 1988.

[28] Sebastien Destercke and Mohamed Sallak. An extension of universal generating function in multi-state systems considering epistemic uncertainties. *IEEE Transactions on Reliability*, 62(2):504–514, June 2013.

[29] Yi Ding, Ming J Zuo, Anatoly Lisnianski, and Zhigang Tian. Fuzzy multi-state systems: general definitions, and performance assessment. *IEEE Transactions on Reliability*, 57(4):589–594, 2008.

[30] Nicolae Dăneţ. Interval analysis - a powerfull trend in numerical analysis. 06 2007.

[31] Didier Dubois. Representation, propagation, and decision issues in risk analysis under incomplete probabilistic information. *Risk Analysis: An International Journal*, 30(3):361–368, 2010.

[32] Paul S Dwyer. Linear computations. 1951.

[33] CR Eckberg. Ws-133b fault tree analysis program plan. Technical report, BOEING CO SEATTLE WA, 1964.

[34] Ingvar Eidhammer, Inge Jonassen, and William R Taylor. Structure comparison and structure patterns. *Journal of Computational Biology*, 7(5):685–716, 2000.

[35] MA El-Damcese and NS Temraz. Availability and reliability measures for multistate system by using markov reward model. *Reliability: Theory & Applications*, 6(3 (22)), 2011.

[36] Eugene C Freuder and Alan K Mackworth. Constraint satisfaction: An emerging paradigm. In *Foundations of Artificial Intelligence*, volume 2, pages 13–27. Elsevier, 2006.

[37] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[38] Boris Gnedenko and Igor A Ushakov. *Probabilistic reliability engineering*. John Wiley & Sons, 1995.

[39] R Goulding, N Jayasuriya, and E Horan. A Bayesian network model to assess the public health risk associated with wet weather sewer overflows discharging into waterways. *Water research*, 46(16):4933–40, October 2012.

[40] Lars Grunske, Peter Lindsay, Egor Bondarev, Yiannis Papadopoulos, and David Parker. An outline of an architecture-based method for optimizing dependability attributes of software-intensive systems. In *Architecting dependable systems IV*, pages 188–209. Springer, 2007.

[41] Michael AS Guth. A probabilistic foundation for vagueness and imprecision in fault-tree analysis. *IEEE Transactions on Reliability*, 40(5):563–571, 1991.

[42] Walter J Gutjahr. Software dependability evaluation based on markov usage models. *Performance evaluation*, 40(4):199–222, 2000.

[43] Ruth M Hall and Christina M Collis. Mobile gene cassettes and integrons: capture and spread of genes by site-specific recombination. *Molecular microbiology*, 15(4):593–600, 1995.

[44] Eldon Hansen and G William Walster. *Global optimization using interval analysis: revised and expanded*, volume 264. CRC Press, 2003.

[45] ER Hansen. Publications related to early interval work of re moore. *Web document: interval. louisiana. edu/Moores early papers/bibliography. html*, 2001.

[46] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, September 1995.

[47] Toshiyuki Inagaki. Interdependence between safety-control policy and multiple-sensor schemes via dempster-shafer theory. *IEEE Transactions on Reliability*, 40(2):182–188, 1991.

[48] Rashmi Jain and George Elias. Exploring attributes for systems architecture evaluation.

[49] Luc Jaulin. *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*, volume 1, chapter Chapter 4. Springer Science & Business Media, 2001.

[50] Mohammad Sadegh Javadi, Azim Nobakht, and Ali Meskarbashee. Fault tree analysis approach in reliability assessment of power system. *International Journal of Multidisciplinary Sciences and Engineering*, 2(6):46–50, 2011.

[51] Arnold Kaufmann, R Cruon, and Daniel Grouchko. Modèles mathématiques pour l'étude de la fiabilité des systèmes. 1975.

[52] Nima Khakzad, Faisal Khan, and Paul Amyotte. Quantitative risk analysis of offshore drilling operations: A Bayesian approach. *Safety Science*, 57:108–117, August 2013.

[53] Shinya Kikuchi and Partha Chakroborty. Place of possibility theory in transportation analysis. *Transportation Research Part B: Methodological*, 40(8):595–615, 2006.

[54] Young H Kim, Kenneth E Case, and PM Ghare. A method for computing complex system reliability. *IEEE Transactions on Reliability*, 21(4):215–219, 1972.

[55] Ar Kimura, R Kasada, N Iwata, H Kishimoto, CH Zhang, J Isselin, P Dou, JH Lee, N Muthukumar, T Okuda, et al. Development of al added high-cr ods steels for fuel cladding of next generation nuclear systems. *Journal of Nuclear Materials*, 417(1-3):176–179, 2011.

[56] Claude Kirchner and Hervé Mathieu. Digital technology at the heart of our lives.

[57] Uffe B. Kjarulff and Anders L. Madsen. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*. Springer-Verlag New York Inc., 2013.

[58] Duane Kritzinger. *Aircraft System Safety: Assessments for Initial Airworthiness Certification*. Woodhead Publishing, 2016.

[59] Way Kuo and V Rajendra Prasad. An annotated overview of system-reliability optimization. *IEEE Transactions on reliability*, 49(2):176–187, 2000.

[60] Way Kuo, V Rajendra Prasad, Frank A Tillman, and Ching-Lai Hwang. *Optimal reliability design: fundamentals and applications*. Cambridge university press, 2001.

[61] Helge Langseth and Luigi Portinale. Bayesian networks in reliability. *Reliability Engineering & System Safety*, 92(1):92–108, January 2007.

[62] JC Laprie, J Arlat, JP Blanquart, A Costes, Y Crouzet, Y Deswarte, JC Fabre, H Guillermain, M Kaâniche, K Kanoun, et al. Guide de la sûreté de fonctionnement, 324 p. *Cépaduès Editions, Toulouse, France*, 1995.

[63] Gregory Levitin et al. *The universal generating function in reliability analysis and optimization*, volume 6. Springer, 2005.

[64] Gregory Levitin, Anatoly Lisnianski, Hanoch Ben-Haim, and David Elmakis. Redundancy optimization for series-parallel multi-state systems. *IEEE Transactions on Reliability*, 47(2):165–172, 1998.

[65] Olivier Lhomme. Consistency techniques for numeric csps. In *IJCAI*, volume 93, pages 232–238, 1993.

[66] Chun-yang Li, Xun Chen, Xiao-shan Yi, and Jun-yong Tao. Interval-valued reliability analysis of multi-state systems. *IEEE Transactions on Reliability*, 60(1):323–330, 2011.

[67] Philipp Limbourg. *Dependability modelling under uncertainty*. Springer, 2008.

[68] Nikolaos Limnios and Dumitru Cezar Ionescu. *Statistical and probabilistic models in reliability*. Springer Science & Business Media, 2012.

[69] Anatoly Lisnianski. The markov reward model for a multi-state system reliability assessment with variable demand. *Quality Technology & Quantitative Management*, 4(2):265–278, 2007.

[70] Anatoly Lisnianski and Ilia Frenkel. Non-homogeneous markov reward model for aging multi-state system under minimal repair. *International Journal of Performability Engineering*, 5(4):303–312, 2009.

[71] Anatoly Lisnianski and Gregory Levitin. *Multi-state system reliability: assessment, optimization and applications*, volume 6. World Scientific Publishing Co Inc, 2003.

[72] Rudolf J Lohner. Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In *Institute of mathematics and its applications conference series*, volume 39, pages 425–425. Oxford University Press, 1992.

[73] Imane Maatouk, Eric Chatelet, and Nazir Chebbo. Cost minimization of maintained multi-state system using optimal load distribution. In *2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering*, pages 1005–1009. IEEE, 2012.

[74] Thomas J Macke, David J Ecker, Robin R Gutell, Daniel Gautheret, David A Case, and Rangarajan Sampath. Rnamotif, an rna secondary structure definition and search algorithm. *Nucleic acids research*, 29(22):4724–4735, 2001.

[75] Manish Malhotra and Kishor S Trivedi. Power-hierarchy of dependability-model types. *IEEE Transactions on Reliability*, 43(3):493–502, 1994.

[76] Marija Mihova and Natasha Maksimova. Estimation of minimal path vectors of multi state two terminal networks with cycles control. *Mathematica Balkanica*, 25(4):437–447, 2011.

[77] Marija Mihova and Nina Synagina. An algorithm for calculating multi-state network reliability using minimal path vectors. 2008.

[78] Krishna B Misra. An algorithm for the reliability evaluation of redundant networks. *IEEE Transactions on Reliability*, 19(4):146–151, 1970.

[79] Ramon E Moore. *Interval analysis*, volume 4. Prentice-Hall Englewood Cliffs, NJ, 1966.

[80] Ramon E Moore. *Methods and applications of interval analysis*. SIAM, 1979.

[81] Ramon E Moore, R Baker Kearfott, and Michael J Cloud. *Introduction to interval analysis*, volume 110. Siam, 2009.

[82] R.E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. October 1962.

[83] Yves Mortureux. La sûreté de fonctionnement: méthodes pour maîtriser les risques. 2001.

[84] Jogesh K Muppala, Manish Malhotra, and Kishor S Trivedi. Markov dependability models of complex systems: Analysis techniques. In *Reliability and Maintenance of Complex Systems*, pages 442–486. Springer, 1996.

[85] Ivan Mura and Andrea Bondavalli. Markov regenerative stochastic petri nets to model and evaluate phased mission systems dependability. *IEEE Transactions on Computers*, (12):1337–1351, 2001.

[86] Daniel Noyes and François Pérès. Analyse des systèmes-sûreté de fonctionnement. *Techniques de l'ingénieur*, 2007.

[87] Alain Pagès and Michel Gondran. *Fiabilité des systèmes*, volume 39. Eyrolles, 1980.

[88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, CA., 1988.

[89] Miodrag Petkovic, Miodrag Petković, Miodrag S Petkovic, and Ljiljana D Petkovic. *Complex interval arithmetic and its applications*, volume 105. John Wiley & Sons, 1998.

[90] C.P. Pfleeger. Data Security. In A. Ralston et Al., editor, *Encyclopedia of Computer Science*, pages 504–507. Nature Publishing Group, 2000.

[91] Siqi Qiu. *Modèles graphiques de l'évaluation de Sûreté de Fonctionnement et l'analyse des risques des Systèmes de Systèmes en présence d'incertitudes*. PhD thesis, Compiègne, 2014.

[92] Siqi Qiu, Mohamed Sallak, Walter Schön, and Zohra Cherfi-Boulanger. Availability assessment of railway signalling systems with uncertainty analysis using statecharts. *Simulation Modelling Practice and Theory*, 47:1–18, 2014.

[93] Jose E Ramirez-Marquez and David W Coit. A monte-carlo simulation approach for approximating multi-state two-terminal reliability. *Reliability Engineering & System Safety*, 87(2):253–264, 2005.

[94] Jose Emmanuel Ramirez-Marquez and David W Coit. Multi-state component criticality analysis for reliability improvement in multi-state systems. *Reliability Engineering & System Safety*, 92(12):1608–1619, 2007.

[95] Kaoutar Rhazali. *Optimisation de la disponibilité des systèmes multi-Etats*. Projet Industriel de Fin d'Etudes, 2015.

[96] Gerardo Rubino and Bruno Sericola. *Markov chains and dependability theory*. Cambridge University Press, 2014.

[97] Mohamed Sallak. *Évaluation de paramètres de sûreté de fonctionnement en présence d'incertitudes et aide à la conception: application aux Systèmes Instrumentés de Sécurité*. PhD thesis, Institut National Polytechnique de Lorraine, 2007.

[98] Mohamed Sallak, Walter Schön, and Felipe Aguirre. The transferable belief model for reliability analysis of systems with data uncertainties and failure dependencies. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 40:266–278, 2010.

[99] Mohamed Sallak, Walter Schön, and Felipe Aguirre. Extended component importance measures considering aleatory and epistemic uncertainties. *IEEE TRANSACTIONS ON RELIABILITY*, pages 1–35, 2012.

[100] Mohamed Sallak, Walter Schön, and Felipe Aguirre. Reliability assessment for multi-state systems under uncertainties based on the dempster–shafer theory. *IIE Transactions*, 45(9):995–1007, 2013.

[101] Mohamed Sallak, Walter Schön, and Felipe Aguirre. Reliability assessment for multi-state systems under uncertainties based on the Dempster-Shafer theory. *IIE Transactions*, page 26, 2013.

[102] Appajosyula Satyanarayana and Mark K Chang. Network reliability and the factoring theorem. *Networks*, 13(1):107–120, 1983.

[103] Christophe Simon and Philippe Weber. Imprecise reliability by evidential networks. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 223(2):119–131, 2009.

[104] Damjan Škulj. Efficient computation of the bounds of continuous time imprecise markov chains. *Applied mathematics and computation*, 250:165–180, 2015.

[105] Alireza Soroudi and Turaj Amraee. Decision making under uncertainty in energy systems: State of the art. *Renewable and Sustainable Energy Reviews*, 28:376–384, 2013.

[106] Rita L. Sousa and Herbert H. Einstein. Risk analysis during tunnel construction using Bayesian Networks: Porto Metro case study. *Tunnelling and Underground Space Technology*, 27(1):86–100, January 2012.

[107] Anthony J Spurgin. *Human reliability assessment theory and practice*. CRC press, 2009.

[108] Teruo Sunaga. Theory of interval algebra and its application to numerical analysis. *RAAG memoirs*, 2(29-46):209, 1958.

[109] José Geraldo Torres-Toledano and Luis Enrique Sucar. Bayesian Networks for Reliability Analysis of Complex Systems. In Helder Coelho, editor, *Progress in Artificial Intelligence - IBERAMIA 98*, page 465. Springer Berlin Heidelberg, 1998.

[110] Matthias Troffaes, Jacob Gledhill, Damjan Škulj, and Simon Blake. Using imprecise continuous time markov chains for assessing the reliability of power networks with common cause failure and non-immediate repair. SIPTA, 2015.

[111] Matthias CM Troffaes, Gero Walter, and Dana Kelly. A robust bayesian approach to modeling epistemic uncertainty in common-cause failure models. *Reliability Engineering & System Safety*, 125:13–21, 2014.

[112] Edward Tsang. *Foundations of constraint satisfaction: the classic text*. BoD–Books on Demand, 2014.

[113] IA Ushakov. A universal generating function. *Soviet Journal of Computer and Systems Sciences*, 24(5):118–129, 1986.

[114] Lev V Utkin and Frank PA Coolen. Imprecise reliability: an introductory overview. In *Computational intelligence in reliability engineering*, pages 261–306. Springer, 2007.

[115] A Villemeur. Sûreté de fonctionnement des systèmes industriels, edition eyrolles, 1997.

[116] Alain Villemeur. Sureté de fonctionnement des systèmes industriels: fiabilité-facteurs humains, informatisation. 1988.

[117] Peter Walley. Imprecise probabilities. 2000.

[118] Hongzhou Wang and Hoang Pham. *Reliability and optimal maintenance*. Springer Science & Business Media, 2006.

[119] Benkt Wangler and Alexander Backlund. Information systems engineering: What is it? In *CAiSE Workshops (2)*, pages 427–437, 2005.

[120] P. Weber, G. Medina-Oliva, C. Simon, and B. Iung. Overview on Bayesian networks applications for dependability, risk analysis and maintenance areas. *Engineering Applications of Artificial Intelligence*, 25(4):671–682, June 2012.

[121] Timothy A Wheeler. Treatment of uncertainties associated with pras in risk-informed decision making (nureg1855). Technical report, Sandia National Laboratories, 2010.

[122] Joon-Eon Yang, Mee-Jung Hwang, Tae-Yong Sung, and Youngho Jin. Application of genetic algorithm for reliability allocation in nuclear power plants. *Reliability Engineering & System Safety*, 65(3):229–238, 1999.

[123] Yixin Ye, Ignacio E Grossmann, Jose M Pinto, and Sivaraman Ramaswamy. Markov chain minlp model for reliability optimization of system design and maintenance. In *Computer Aided Chemical Engineering*, volume 44, pages 1483–1488. Elsevier, 2018.

[124] Gu Yingkui and Li Jing. Multi-state system reliability: A new and systematic review. *Procedia Engineering*, 29:531–536, 2012.

[125] Rosalind Cecily Young. The algebra of many-valued quantities. *Mathematische Annalen*, 104(1):260–290, 1931.

[126] Haiyang Yu, Chengbin Chu, and Éric Châtelet. Availability optimization of a redundant system through dependency modeling. *Applied Mathematical Modelling*, 38(19-20):4574–4585, 2014.

[127] Haiyang Yu, Farouk Yalaoui, Ėric Châtelet, and Chengbin Chu. Optimal design of a maintainable cold-standby system. *Reliability Engineering & System Safety*, 92(1):85–91, 2007.

[128] D. Zhang, X.P. Yan, Z.L. Yang, A. Wall, and J. Wang. Incorporation of formal safety assessment and Bayesian network in navigational risk estimation of the Yangtze River. *Reliability Engineering & System Safety*, 118:93–105, October 2013.

[129] Hao Zhang and Michael Beer. Interval quasi-monte carlo method for reliability assessment with imprecise probability. 2012.

[130] Xiaopin Zhong, Mohamed Ichchou, and Alexandre Saidi. Reliability assessment of complex mechatronic systems using a modified nonparametric belief propagation algorithm. *Reliability engineering & system safety*, 95(11):1174–1185, 2010.

[131] Enrico Zio, Marco Marella, and Luca Podofillini. A monte carlo simulation approach to the availability assessment of multi-state systems with operational dependencies. *Reliability Engineering & System Safety*, 92(7):871–882, 2007.

[132] Enrico Zio, Luca Podofillini, and Gregory Levitin. Estimation of the importance measures of multi-state elements by monte carlo simulation. *Reliability Engineering & System Safety*, 86(3):191–204, 2004.

[133] Gilles Zwingelstein. La maintenance basée sur la fiabilité. *Hermès*, 199, 1996.