REBAIAIA MOHAMED-LARBI

A CONTRIBUTION TO THE EVALUATION AND OPTIMIZATION OF NETWORKS RELIABILITY

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en génie mécanique
pour l'obtention du grade de Philosophiae Doctor (Ph.D.)

DÉPARTEMENT DE GÉNIE MÉCANIQUE
FACULTÉ DES SCIENCES ET GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

2011

Résumé

L'évaluation de la fiabilité des réseaux est un problème combinatoire très complexe qui nécessite des moyens de calcul très puissants. Plusieurs méthodes ont été proposées dans la littérature pour apporter des solutions. Certaines ont été programmées dont notamment les méthodes d'énumération des ensembles minimaux et la factorisation, et d'autres sont restées à l'état de simples théories.

Cette thèse traite le cas de l'évaluation et l'optimisation de la fiabilité des réseaux. Plusieurs problèmes ont été abordés dont notamment la mise au point d'une méthodologie pour la modélisation des réseaux en vue de l'évaluation de leur fiabilités. Cette méthodologie a été validée dans le cadre d'un réseau de radio communication étendu implanté récemment pour couvrir les besoins de toute la province québécoise. Plusieurs algorithmes ont aussi été établis pour générer les chemins et les coupes minimales pour un réseau donné. La génération des chemins et des coupes constitue une contribution importante dans le processus d'évaluation et d'optimisation de la fiabilité. Ces algorithmes ont permis de traiter de manière rapide et efficace plusieurs réseaux tests ainsi que le réseau de radio communication provincial. Ils ont été par la suite exploités pour évaluer la fiabilité grâce à une méthode basée sur les diagrammes de décision binaire. Plusieurs contributions théoriques ont aussi permis de mettre en place une solution exacte de la fiabilité des réseaux stochastiques imparfaits dans le cadre des méthodes de factorisation. A partir de cette recherche plusieurs outils ont été programmés pour évaluer et optimiser la fiabilité des réseaux. Les résultats obtenus montrent clairement un gain significatif en temps d'exécution et en espace de mémoire utilisé par rapport à beaucoup d'autres implémentations.

Mots-clés: *Fiabilité, réseaux, optimisation, diagrammes de décision binaire, ensembles des chemins et coupes minimales, algorithmes, indicateur de Birnbaum, systèmes de radio télécommunication, programmes.*

Abstract

Efficient computation of systems reliability is required in many sensitive networks. Despite the increased efficiency of computers and the proliferation of algorithms, the problem of finding good and quickly solutions in the case of large systems remains open. Recently, efficient computation techniques have been recognized as significant advances to solve the problem during a reasonable period of time. However, they are applicable to a special category of networks and more efforts still necessary to generalize a unified method giving exact solution.

Assessing the reliability of networks is a very complex combinatorial problem which requires powerful computing resources. Several methods have been proposed in the literature. Some have been implemented including minimal sets enumeration and factoring methods, and others remained as simple theories.

This thesis treats the case of networks reliability evaluation and optimization. Several issues were discussed including the development of a methodology for modeling networks and evaluating their reliabilities. This methodology was validated as part of a radio communication network project. In this work, some algorithms have been developed to generate minimal paths and cuts for a given network. The generation of paths and cuts is an important contribution in the process of networks reliability and optimization. These algorithms have been subsequently used to assess reliability by a method based on binary decision diagrams. Several theoretical contributions have been proposed and helped to establish an exact solution of the stochastic networks reliability in which edges and nodes are subject to failure using factoring decomposition theorem. From this research activity, several tools have been implemented and results clearly show a significant gain in time execution and memory space used by comparison to many other implementations.

Key-words: *Reliability, Networks, optimization, binary decision diagrams, minimal paths set and cuts set, algorithms, Birnbaum performance index, Networks, radio-telecommunication systems, programs.*

Preface

This thesis has been prepared under the supervision of Dr. Daoud Ait-Kadi, Professor at the Mechanical Engineering Department of Laval University- Quebec, Canada.

The thesis has been organized in the form of articles insertion thesis. The activities related to this thesis have been carried out at the department of mechanical engineering and at the direction of information technologies and telecommunication and precisely at the national interoperable radio communication network services.

The thesis includes six articles, co-authored in all of them by Pr. Daoud Ait-Kadi and in two by my colleague Arturo Merlano. I have acted in all of them as the first author and performed all the mathematical models, coding and algorithms implementation, analysis and validation of the results, as well as writing all the drafts of the articles. The paper that deals with the availability optimization has been written by Alain Ratle and co-authored by Pr. Daoud Ait-Kadi and I acted as the third author. It has been decided that this article will not appear in the list of articles presented in this thesis.

The content of the articles list is as follows:

1 Mohamed-larbi Rebaiaia, Daoud Ait-Kadi, Arturo Merlano, A Practical Algorithm for network reliability evaluation based on the factoring theorem — A case study of A generic Radio-communication network, Journal of Quality, vol.16, N0 5, November 2009.

2 Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, Arturo Merlano, A Methodology for Modeling and evaluating the Reliability of a Radio communication Network, the 1st IFAC Workshop on "Advanced Maintenance Engineering, Services and Technology", IFAC-PapersOnLine.net, ISSN: 1474-66, 2010.

3 Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, An efficient Algorithm for Enumerating Minimal PathSets in Networks, submitted to the Journal of Reliability Engineering

& System Safety Published by Elsevier, 2011.

4 Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, Algorithms for Generating Minimal Cutsets from Binary Decision Diagrams, submitted to the European Journal of Operational Research, 2011.

5 Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, A Contribution for Computing the Reliability of Networks using Reduced Binary Decision, submitted to the European Journal of Operational Research, 2011.

6 Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, Méthode de factorisation polygone-à chaine pour l'évaluation exacte de la fiabilité des réseaux dont les nœuds et les liens sont imparfaits, submitted to the Journal Européen des Systèmes Automatisés, 2011.

7 Ratle Alain., Ait-Kadi Daoud, and Rebaiaia Mohamed-Larbi, Evolutionary Algorithm for availability Optimization of Parallel/Series Systems, submitted to the Journal of Reliability Engineering & System Safety Published by Elsevier, 2011.

In memory of my mother Larem and my father Ahmed.
To my dearest wife Saliha and my children,
Anis, Islam, Ahmed-Rayan and Taha-Assil.
To my sister and my brothers.

Acknowledgements

First of all, I would like to express my deepest appreciation to my advisor, Professor Daoud Ait-Kadi, for his guidance and supervision that made the completion of this research possible. I also extend my sincere appreciation to each of the committee members' examination of the project of this dissertation, Dr. Nour El-Fath Mustapha and Dr. Djamali Anouar, Professors in mechanical engineering.

I want to thank everyone who helped me directly or indirectly in the preparation of the research contained in this thesis. Especially, I'm grateful to my friends and colleagues studying in the mechanical engineering department and those working as professional of research at the FOR@C consortium.

Finally, on a more personnel note, I wish to express my gratefulness to my wonderful wife for her love, sacrifice, patience, encouragement and support.

Table of Contents

List of Tables

List of figures

Chapter 1

General Introduction

1.1 INTRODUCTION

Nowadays, sensitive areas are designed into systems, sometimes homogeneous and sometimes heterogeneous, so more complexities. It appears clear that, at the design phase of the life cycle, the study should precisely focus on the system survivability aspects. In other words, the system must be robust, reliable, secure, and especially extensible, open and must survive to crash.

Many social, economic, services and manufacturing systems can be modeled by a network. In many such networks, the physical problem has sources and sinks nodes and a network of links connecting all the nodes. In other words, given a stochastic network ***N*** modeling a physical system in which each edge and/or each node can fail statistically independently, with a known probability. The network reliability analysis problem consists of measuring network reliability given failure probabilities for the edges/nodes. Typical reliability measures could be presented as the probability *Pr*{there exist operating paths from a node *s* to each node in a subset ***K***} or more simply the probability *Pr*{there exist at least an operating path from a source node *s* to a sink node *t* in the network}. Computing such measure has been proved to be NP-hard. Rosenthal (1973) was the first to show that the recognition problem of determining if a network contains reliable Steiner Tree of given cardinality is NP-Complete. In practical cases, network reliability algorithms belong to the class of NP-hard problems, and in the theory they have been classified as #P-complete by Valiant (Valian, 1979).

When addressing the problem of evaluating networks reliability for a given system whose topological structure is known, we are quickly confronted with two issues requiring answers beforehand and which are to ask: 1- is the reliability of all components available, 2- what are the appropriate tools for computing the reliability in accordance to network dimensions.

In this thesis, we provide clear and practical answers to the earlier problems. Indeed in this presentation, several solutions have been introduced. The first answer detailed in chapter 2, is to propose a framework and a unified approach to determine all parameters that calculate the reliability whatever the component characteristics (electric, electronic, mechanical or simply software). The second answer is recorded in all the rest of this dissertation from chapter 3 to chapter 7, in which we have proposed algorithms based solution and data structures based coding and manipulation solution. The principle of the algorithms is to give flexible solutions. It is noted that chapter 1 to chapter 7 reflect the contents of articles proposed for publication where two of them have been already appeared.

The following sections of the present chapter are as follows:

Section 2 introduces some important definitions and methods for computing the reliability. Section 3 proposes a concise state-of-the-art. The thesis objectives and a review of the literature are presented in section 4 and section 5. In section 6, we review the contributions and details of each one of the following chapters. Chapter 1 concludes in section 7.

1.2 Background Preliminaries

1.2.1 Definitions

1.2.1.1 Stochastic graph

A stochastic graph $G = (V, E)$ is a finite set V of nodes and a finite set E of incidence relations on the nodes called edges. The edges are considered as transferring a commodity between nodes with a probability p. They may be directed or undirected and are weighted by their existence probabilities. The graph in such case, models a physical network, which represents a linked set of components providing services. In this work, other terms are used to define stochastic graphs such as reliability model or simply *network*. They give exactly the same meaning.

1.2.1.2 Subgraph and partial graph

A subgraph of a given graph $G = (V, E)$ is a graph $G' = (V', E')$ such that $V' \subset V$ and $E' = (V' \times V') \cap E$.

A partial graph of a given graph $G = (V, E)$ is a graph $G'' = (V, E'')$ such that $E'' \subset E$.

1.2.1.3 Graph state, associated probability and associated partial graph

As each element of the network during system operations may be up (operating) or may be down (failure), thus a state Boolean cardinality $element\ states = up, down = 2$. There are 2^{n+m}possible states for a network, with $n = |V|$ and $m = |E|$.

1.2.1.4 Network Reliability

There are different notions of network reliability, i.e, deterministic and stochastic (Frank and Friech, 1971), (Hwang, Tillman and Lee, 1981). Networks can be defined as a physical or organisational infrastructure that can be modeled as a graph composed of nodes and links (directed or undirected) in which each edge has associated value which corresponds to the probability that such component is functioning. Because each edge and each node can fail with a probability value, the reliability of a network $G = (V, E)$ is defined as the probability that the system (network) will perform its intended functions without failure over a given period of time and under specific conditions.

1. 2.1.5 Paths, chains, connected graph, cuts, minimal paths, minimal cuts

1- A path is a chain $\mu = (x_1, \ldots, x_q)$ in which the terminal endpoint of arc μ_i is the initial endpoint of arc μ_{i+1} for all $i < q$. Hence, we often write $\mu = (x_1, x_{k+1})$, where k is the number of edges and k is considered as the length of the path (chain).

2- A graph G is said to be *connected* if between any two nodes $x, y \in V$ there exists a chain $\mu = (x, y)$.

3- A path P in a graph G is said to be a *1*-path if any two nodes $x, y \in V$ they are linked by only one edge. Any path in a graph G which is a *1*-path is said to be a branch.

4- Minpath: A subset of a path with minimal number of elements that still make the system functioning.

5- Minpath set (MPS): The set of all minpaths of a network.

6- A cut C is a set of elements such that if all of them are true then the system is failed.

7- Mincut: A subset of a cut with minimal number of components that still make the system fail.

8- Mincut set (MCS): The set of all mincuts.

Let x_i be the state component and x the state vector, they can define what follows:

- $x_i(t) = \begin{cases} 1 & if \text{ the component is up at time } t \\ 0 & otherwise \end{cases}$

- $x = (x_1, x_2, \dots, x_m)$ a state vector of the system S of order m such that $x \in \Omega_i = \{0,1\}^m$ the state space of the system.

The system is then represented by its structure function $\Phi\colon \Omega \to \{0,1\}$ defined as follows :

$$\Phi(x) = \begin{cases} 1 & if \text{ the system is up when the state vector is } x \\ 0 & if \text{ the system is down when the state vector is } x \end{cases}$$

After specifying the structure function Φ, a probabilistic structure is defined. The usual framework is to assume that the state of the i^{th} component is a random binary variable and that the state vectors are independent..

If a system contains P minpath set $P_1, P_2, \dots, P_P$ and C mincut set $C_1, C_2, \dots, C_C$ its structure function can be represented by :

$$\Phi(x) = \max_{1 \le j \le P} \min_{i \in P_j} x_i = \min_{1 \le j \le C} \max_{i \in C_j} x_i$$

The basic expression of the reliability R of a network S is presented in the following form :

$$R(S) = Probability(S \text{ non} - \text{defaulting on the time interval } [0,t[) = Pr\{\Phi(X) = 1\}$$
$$= E\{\Phi(X)\} = \sum_{X \in \Omega_i} \Phi(X) Pr\{X = x\}$$

where $E\{\Phi(X)\}$ is the mathematical expectation, and $p_i = Pr\{X_i = 1\}$ and $q_i = Pr\{X_i = 0\} = 1 - p_i$.

Note that p_i is called elementary reliability. We can observe that this is a static problem, because time is not explicitly used in the analysis.

However in case the network where nodes are also prone to failure it is possible to replace nodes by arcs and thus we return to a network without imperfect nodes. In such case the reliability is expressed by:

$$R(G) = \prod_{v_j \in K} p_i \, . R(G')$$

where G' is the graph G with perfect terminal nodes and K is the set of perfect nodes of the network.

1.2.2 Survey of some existing methods for networks reliability evaluation

Network reliability evaluation is a difficult problem. There are several approaches to tackle it. For any practical problem of significant size, one must use a computational program. In the literature many articles talk about exact methods such as enumeration methods which are divided into 2 classes: enumeration of states and enumeration of paths and cuts and approximating methods such as simulation. Generally, these methods are used as a pre-computation for others methods such as the Sum of Disjoint Product (SDP) methods. In the following we present some of them.

1.2.2.1 Enumeration methods

1.2.2.1.1 State-Space Enumeration

State-space enumeration method is the most direct brute force approach for computing the reliability of a network. It proceeds simply by determining the whole set of state vectors, checking for each one if the network is operational or not. The whole set of state vectors represents all the combinations where each of the *m* edges can be good or bad, resulting in 2^m combinations. Each of these combinations is considered as an event E_i. These events are all mutually exclusive (disjoint) and the reliability expression is simply the probability of the union of the subset of events that contain a path between *s* and *t* which is expressed as follows:

$$\mathrm{R_{s,t}}(\mathrm{S}) = \Pr(E_1 \cup E_2 \cup \ldots \cup E_m) = \Pr(E_1) + \Pr(E_2) + \cdots + P(E_m) \qquad (1.1)$$

where $E_1 \cup E_2 = \emptyset \; \forall i, \forall j, i \neq j$

Suppose we want to use state enumeration method to evaluate the reliability between node *a* and node *c* of the network presented in figure 1.1. First, we adopt some conventional terms. Let the term *good* means that there is at least one path from *a* to *c* for the given combination of good and failed edges. The term *bad*, on the other hand, means that there are no paths from a to c for the given combination of good and failed edges. The result-good or bad is determined by inspection of the graph, they are reported in table 1.1.

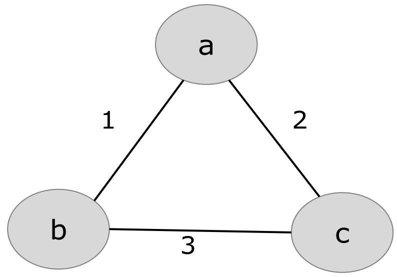


Figure 1.1 A simple network

1- No failure: $E_1 = 1\ 2\ 3$ Good	# Combination: $\binom{3}{0} = \frac{3!}{0!3!} =$ 1
2- One failure: $E_2 = 1'\ 2\ 3$; $E_3 = 1\ 2'\ 3$; $E_4 = 1\ 2\ 3'$ Good Good Good	# Combination: $\binom{3}{1} = \frac{3!}{1!2!} =$ 3
3- Two failures: $E_5 = 1'\ 2'\ 3$; $E_6 = 1\ 2'\ 3'$; $E_7 = 1'\ 2\ 3'$ Bad Bad Good	# Combination: $\binom{3}{2} = \frac{3!}{2!1!} =$ 3
4- Three failures: $E_8 = 1'\ 2'\ 3'$ Bad	# Combination: $\binom{3}{3} = \frac{3!}{3!0!} =$ 1

The reliability is deduced from the addition of the good events. It is as follows :

$$R_{a,c}(G) = [p_1p_2p_3] + [q_1p_2p_3 + p_1q_2p_3 + p_1p_2q_3] + [q_1p_2q_3] = p_2 + p_1p_3 - p_1p_2p_3$$

1.2.2.1.2 Path enumeration – Cut enumeration

This method is executed in two steps. First step consists of enumerating MPS or MCS. In second step the reliability evaluation needs the development of the symbolic expression in terms of the probability of various components being operational/non operational. If MPS/MCS are mutually exclusive, the probability of the union of m events (corresponding to components state; working/failed) can be written if MPS = $\{P_1, P_2, \ldots, P_m\}$ and MCS = $\{C_1, C_2, \ldots, C_m\}$ where C_j represents the event "the components of the jth minimal cut are not functioning", thus

$$= 1 - (\Pr(C_1) + \Pr(C_2) + \cdots + \Pr(C_n))$$

$$R(G) = \Pr(P_1 \cup \ldots \cup P_m) = \Pr(P_1) + \Pr(P_2) + \cdots + P(P_m) \,/ E_1 \cup E_2 = \emptyset\ \forall i, \forall j, i \neq j \qquad (1.2)$$

For the same example presented earlier (figure 1.1.), the algorithm generates two minpaths,

$Path_1$: {1,3}; $Path_2$: {2}.

The structure function is equal to: $\Phi(X) = 1 - (1 - x_1x_3)(1 - x_2) = x_2 + x_1x_3 - x_1x_2x_3$

and the reliability is :

$R_{a,c}(G) = E\{\Phi(X)\} = p_2 + p_1p_3 - p_1p_2p_3$

If $p_1 = 0.9$; $p_2 = 0.9$; $p_3 = 0.9$, then $R_{a,c}(G) = 0.981$

Note that state enumeration and path enumeration methods give the same expression thus the same value of network reliability.

1.2.2.2 Sum of Disjoint Product Methods

1.2.2.2.1 Introduction

The starting point of an analysis is Boolean polynomial expression **B** for system success as a logical sum of the MPS. Dually, the polynomial for failure is a sum of MCS.

Let the system have *m* MPS $P_1, P_2, \dots, P_m$, each minpath is a term of the minimized for of **B**. Due to the s-coherence properties (Barlow and Proschan, 1975), every MPS has all 1-valued variable. The minimal expression for success is:

$$\Phi\ X\ = P_1 + P_2 + \cdots + P_m = P_1 + P_1P_2 + P_1P_2P_3 + \cdots + P_1P_2P_3 \dots P_m \qquad (1.3)$$

The statement is simple to be proved by induction on the Boolean terms due to the following Boolean equality:

$$x + y = x + xy$$

1.2.2.2.2 Inclusion-exclusion formula

The inclusion-inclusion formula also called Poincaré-formula or Poincaré theorem can be used for generating directly the expression of the reliability. It is as follows:

$$E\{\Phi(G)\} = \sum_{1 \le i \le m} E(P_i) - \sum_{1 \le i_1 < i_2 \le m} E(P_{i_1}.P_{i_2}) + \cdots + (-1)^{m+1}.E(P_1.P_2.P_3 \cdots P_m) \quad (1.4)$$

The structure function relative to the early network is : $\Phi(G) = P_1 + P_2$: P_1 and P_2 are the paths. Thus, the reliability is,

$$R\ \ G\ \ = E\{\Phi\ \ G\ \ \} = E\{P_1 + P_2\} = E\{P_1\} + E\{P_2\} - E\{P_1.P_2\} = E\{1{,}3\} + E\{2\} - E\{\{1{,}3\}.\{2\}\}$$
$$= p_2 + p_1.p_3 - p_1 p_2 p_3$$

Note that Poincaré formula generates the same expression as earlier.

It is adequate for the calculus to derive from Poincaré formula a recursive function. It is as follows:

$$R_j = \Pr\left\{\bigcup_{i=1}^{j-1} S_i\right\} = R_j + P_j - \Pr\left\{S_j \bigcap \left(\bigcup_{i=1}^{j-1} S_i\right)\right\} \quad (1.5)$$

R_j : is the term of the reliability at j^{th} step.

1.2.2.2.3 Recursive Disjoint Product (Abraham method)

Recursive disjoint product has been introduced by Abraham (Abraham, 1979). It accomplishes the same objective as Poincaré method but results in a different form of the probability polynomial (Locks, 1980). Given the list of MPSs corresponding to a network. The algorithm builds recursively the expression of the reliability by accumulating the probability of the MPS P, one MPS at a time. The recursion formula of Abraham is :

$$R_j = R_j + \Pr\left\{S_j \bigcap \left(\bigcup_{i=1}^{j-1} S_i\right)\right\} = R_j + \Pr\{S_j \cap \bar{S}_1 \cdots \cap \bar{S}_{j-1}\}$$

For the network in figure 1.1, the method of Abraham is used as follows:

$Path_1$: $\{1,3\} = x_1 x_3$; $Path_2$: $\{2\} = x_2$

The first term corresponds to the first path : Term 1 : $x_1 x_3$;

Outer loop 1:

Term 2: $x_1 x_2$;

Term 3: $x_1 x_3 x_2$;

Stop.

$\Phi(X) = x_1x_3 + \bar{x}_1x_2 + x_1\bar{x}_3x_2;$

Simple checking :

The substitution of the complemented variable in $\Phi \ X$, gives :

$\Phi(X) = x_1x_3 + \bar{x}_1x_2 + x_1\bar{x}_3x_2 = x_1x_3 + (1 - x_1)x_2 + x_1(1 - x_3)x_2 = x_2 + x_1.x_3 - x_1x_2x_3$

and the reliability is :

$R \ G \ = p_2 + p_1.p_3 - p_1p_2p_3$

Proof of the equivalence of Recursive disjoint products and Recursive inclusion-exclusion:

$$P_j = \Pr\{S_j\} = Pr\left\{S_j \bigcap \left(\bigcup_{i=1}^{j-1} S_i\right)\right\} + Pr\left\{S_j \bigcap \left(\overline{\bigcup_{i=1}^{j-1} S_i}\right)\right\}$$

Thus

$$P_j - Pr\left\{S_j \bigcap \left(\bigcup_{i=1}^{j-1} S_i\right)\right\} = Pr\left\{S_j \bigcap \left(\overline{\bigcup_{i=1}^{j-1} S_i}\right)\right\}$$

The accumulated probability is exactly the sum of the probability of the terms.

1.2.2.2.4 Disjoint Products (Heidtmann method, 1989)

This algorithm is a modification of the algorithm given earlier by Abraham. The inversion use multiple-variable. It is simpler and more efficient than Abraham's. On the example of figure 1.1, it proceeds as follows:

First term is : x_1x_3 ;

Second term is : $x_1x_3x_2$

$$\Phi(X) = x_1x_3 + \overline{x_1x_3}x_2 = x_1x_3 + (1 - x_1x_3)x_2 = x_2 + x_1.x_3 - x_1x_2x_3$$

1.2.2.3 Reduction and factorisation

1.2.2.3.1 Reduction rules

In order to reduce the size of network which leads to minimizing the computing cost of the network reliability, it is needed to apply some reduction techniques. The idea behind the reduction is to transform each graph partition into a simplified form, while preserving its reliability. They are (reductions) similar to those of the factoring theorem, which consist of the replacement of a particular structure (e.g. a polygon) embedded in the graph within the abstraction of the rest of the graph. The demonstration of such procedure uses the following reduction rules which are resumed in what follows:

Let $e_a = (u, v)$ and $e_b = (u, w)$ be two series edges in G_K such that degree(v) = 2 and $v \notin K$. Applying reduction procedure leads to obtain the sub-graph G' by replacing e_a and e_b with a single edge $e_c = (u, w)$ and the corresponding reliability is computed by $p_c = p_a p_b$, and it defines Ω = 1 and $K' = K$.

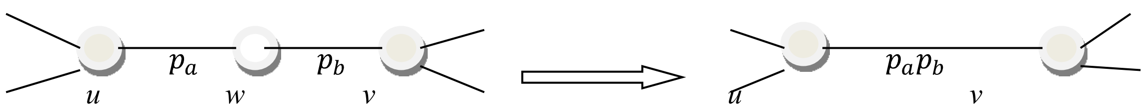


Figure 1.2. Series reduction

Parallel reduction. Let $e_a = (u, v)$ and $e_b = (u, v)$ be two parallel edges in G_K (the network graph) and suppose that $p_i = 1 - q_i$ ($i = a$ or b). A parallel reduction obtains G' by replacing e_a and e_b with single edge $e_c = (u, v)$ with reliability $p_c = (1 - q_a q_b)$, and it defines Ω = 1 and $K' = K$. We note that Ω is a multiplicative operator derived from $R(G_K) = \Omega.R(G'_{K'})$

.

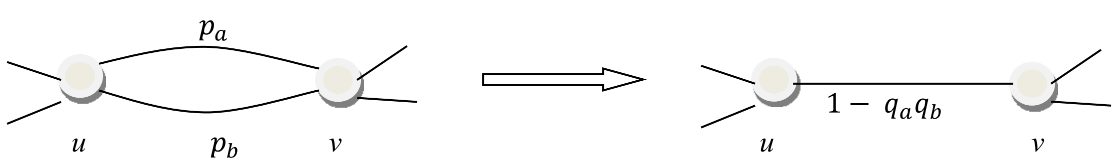


Figure 1.3. Parallel reduction

Let $e_a = (u, v)$ and $e_b = (u, w)$ be two series edges in G_K such that degree(v) = 2 and $u, v, w \in K$. A degree-two reduction obtains G' by replacing e_a and e_b with single edge $e_c = (u, w)$ with reliability ,
$p_c = \frac{p_a p_b}{1 - q_a q_b}$ and it defines Ω = 1 and $K' = K - v$.

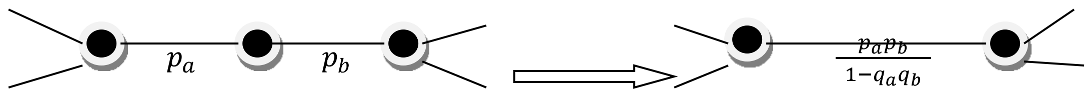


Figure 1.4. Degree two reduction

For directed networks, some additional reduction rules have been introduced (see Deo and Medidi, 1992). They can be presented as follows:

1. All edges going into the source and all edges going out of the sink can be removed. of the sink can be removed. These edges do not lie on any simple source-to-sink path and are thus irrelevant.
2. Every vertex, except the source and the sink, with *0* in-degree or *0* out-degree can be removed.
3. If a single edge is directed into or out of a vertex, its anti-parallel edge can be removed. Since any simple path through this vertex has to use this single edge, the anti-parallel edge is irrelevant.
4. If there is a single edge out of the source or into the sink, then this edge can be contracted. To get the reliability of the original network, the reliability of the reduced network is multiplied by the success probability of the contracted edge.
5. Series edges can be reduced as shown in figure l.2. (Exceptionally the edges are directed from *u* to *v*)
6. Parallel edges can be reduced as shown in figure l.3. (edges are directed from *u* to *v*).
7. Generalized series reduction, analogous to the elimination of degree-2 vertex in undirected networks, can be performed as shown in the following figure.

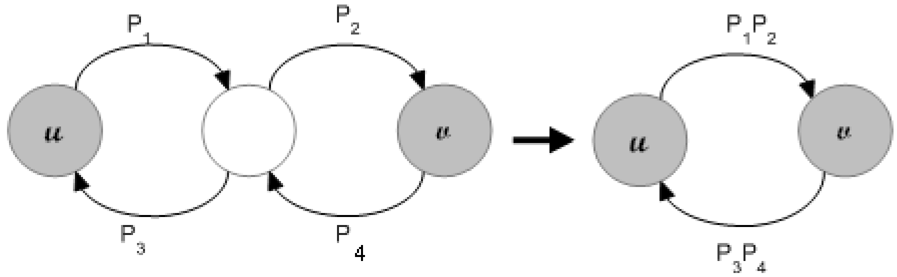


Figure 1.5. Generalized series reduction

Delta-to star reduction. Delta-to star reduction consists of replacing a topological delta structure by a star structure. Nodes of the delta structure must all be *K*-nodes. The added node is not a *K*-node and its probability is computed as follows :

$$p_x = \frac{\alpha}{\alpha+\beta_1}, \quad p_y = \frac{\alpha}{\alpha+\beta_2}\ , \ p_z = \frac{\alpha}{\alpha+\beta_3} \text{ and } \quad p_{u_0} = \frac{(\alpha+\beta_1)(\alpha+\beta_2)(\alpha+\beta_3)}{\alpha^2}$$

Avec $\alpha = p_a p_b + p_a p_c + p_b p_c - 2p_a p_b p_c$ and $\beta_1 = q_a q_b p_c\ , \beta_2 = q_a p_b p_c\ , \beta_3 = p_a q_b q_c$

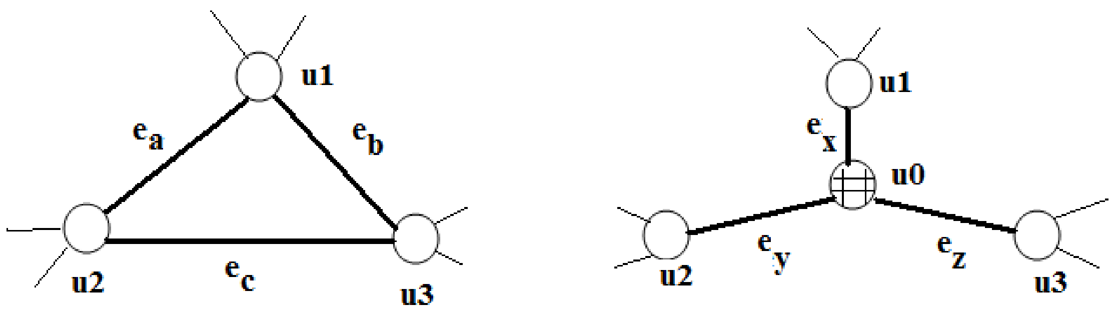


Figure 1.6. Delta-to star reduction

1.2.2.3.2 Generalized factoring theorem

The factoring theorem consists of pivoting on every edge e_i in the graph and decomposing the original problem with respect to two possible states of edge e_i:

$$R(S) = p_i R(S|e_i\, works) + q_i R(S|e_i\, fails)$$

where $R(S)$ is the reliability of the system S and $R(S|e_i works)$ is the reliability of the system S when the edge e_i is in operation and $R(S|e_i fails)$ is the reliability of the system S when the edge e_i is not in operation and each probability p_i is asserted to the edge i and $q_i = (1 - p_i)$ the opposite of p_i. The earlier equation can be recursively applied to the induced graph, until the generated subgraph contains just one edge. Generally, the factoring theorem and reduction rules work together for obtaining the minimal reliability expression possible. Figure 1.6, gives an overview of the application of factoring decomposition using the graph in figure 1.1. In this example, edge 1 has been picked for generating two state-subgraph at left where it has been supposed that edge 1 is working, and subgraph at right the opposite case (edge 1 fails).

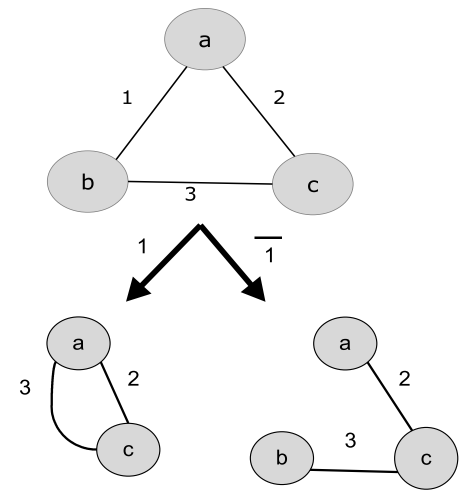


Figure 1.7. First step performing the factoring decomposition.

1.2.3 Binary Decision Diagram (BDD)

Binary Decision Diagrams (BDDs for short) is a clever and simple representation of Boolean expressions. It can be considered as a flexible and dynamic notation of directed acyclic graphs (DAG for short). The idea behind BDDs is their transformation from a DAGs to a more reduced one called ROBDD for Reduced Binary Decision Diagrams(Bryant (1986)). They have received a lot of attention in different fields like computational logics, hardware/software verification and in VLSI design. The implementation and manipulation of BDD algorithms is composed by three procedures introduced in Bryant (1986): *restrict*, *apply* and If-Then-Else (ITE).

The representation and the simplification of a Boolean expression proceeds in 4-steps:

- Construct the binary decision tree (BDT) associated with the graph formula.
- Transform the BDT to a BDD by applying the following rules by :

  a- Merging equivalent leaves of a binary decision tree.

  b- Merging isomorphic nodes.

c- Elimination of redundant tests.

Figure 1.8 gives an overview of BDDs representation and the transformation from a DAG to a ROBDD.

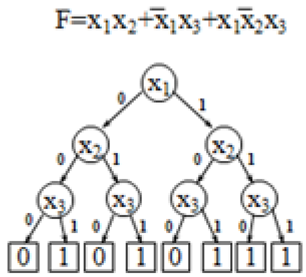
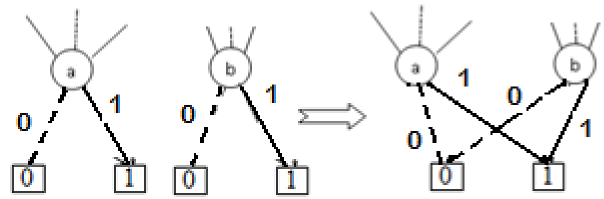
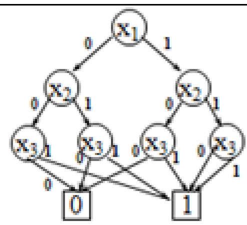
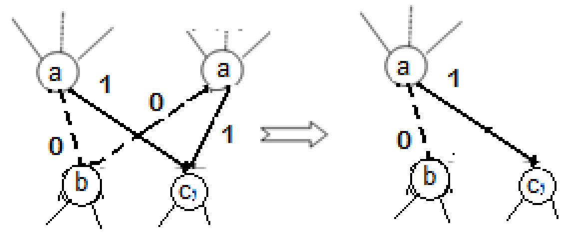
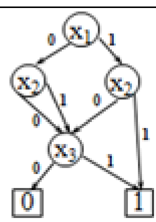
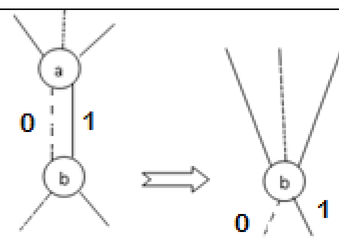
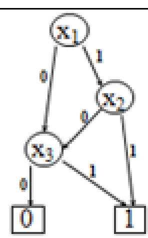
Steps	BDD at the beginning of reduction steps	Reduction rules
Step 1		
Step 2		
Step 3		
		

Figure 1.8. A logic expression and its transformation from a DAG until Normal form (ROBDD).

Network reliability is calculated using the following relation and algorithm and figure 1.9 shows how the algorithm proceeds:

$$P(F) = P\big(ite(x, F_1, F_2)\big) = P(x.F_1) + P(\bar{x}.F_2) = (1 - p_x)P(F_1) + p_xP(F_2)$$

$$= P(F_1) + p_x(P(F_2) - P(F_1))$$

```
Algorithm Reliability_Evaluation(F, G)
    if ( F == 0)
        return 1    //***  Boolean value 1 (one)  ***//
      else  if  (F == 1)
            return 0    //***  Boolean value 0 (zero)  ***//
            else  if (computed-table has entry {F, P_F})
                return  P_F
              else
              P_F = Prob(F1)  +  P(x) * (Prob(F2)-  Prob(F1)  )
              end
          end
       Insert_computed_table ({F, P_F})
       return  P_F
    end
 end
```

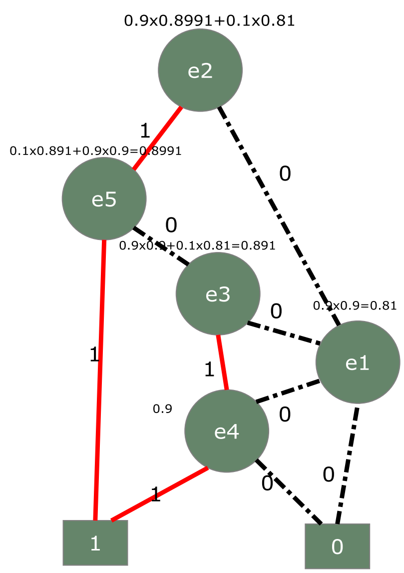


Figure 1.9. Solution given by the earlier algorithm

1.2.4 Reliability optimization

Modern systems are by nature very complex. To remain competitive, the guarantee of high system reliability at low cost is essential. Computing system reliability is usually not sufficient because it would also provide mechanisms to optimize the reliability taking into account budgetary constraints and parameters which could vary in real-time. Several solutions have been published in the literature (see Kuo *et al.*, 2000). Thus, one solution for improving the reliability of a system is to add identical components which can be chosen as design alternatives or by giving more capacity to those that already exist. This model reflects precisely the problem of redundancy allocation. Another method consists of using Binary Decision Diagrams which provides a solution to compute the reliability of a system. The algorithm generates the BDD corresponding to constraints and objective function. The manipulation using APPLY procedure (already cited) consists of composing all the constraints with the objective function. The last BDD will represent the solution of the problem. Another alternative solution to optimizing the reliability of networks is precisely to act on the most important component which can bring some improvements to the reliability. The method is called: the Birnbaum's importance index (Birnbaum, 1969). The solution is simple to be evaluated; it suffices to differentiate the expression of the reliability for each pivoting component. The choice is then fixed on the component which has the highest reliability value. The process is iterated until the expected network reliability is approached.

The discussed three models are presented in what follows:

1.2.4.1 Series-parallel configuration

Consider the following redundancy allocation problem in *m*-stage series system and where in each stage (subsystem) there is a number of components in parallel. The problem can be presented as follows:

$$\max_x R(x) = \prod_{j=1}^{m} R(x_j) = R(x_1) \times R(x_2) \times \ldots \times R(x_m)$$

Subject to:

$$\left| \begin{array}{l} \sum_i \sum_j c_{ij} x_{ij} \leq C \\ \sum_i \sum_j w_{ij} x_{ij} \leq W \\ \qquad x_{ij} \in N \\ \text{where} \\ R(x_1) = 1 - \prod_{j=1}^{m1} (1 - r_{1j})^{x_{1j}}, \\ R(x_2) = 1 - \prod_{j=1}^{m2} (1 - r_{2j})^{x_{2j}}, \\ \qquad \vdots \\ R(x_s) = 1 - \prod_{j=1}^{ms} (1 - r_{1j})^{x_{sj}} \end{array} \right.$$

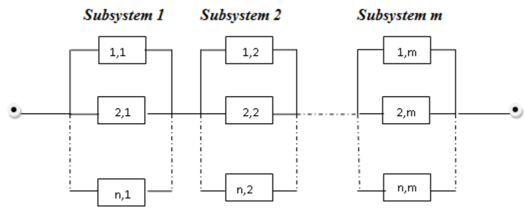


Figure 1.10. A series-parallel system

It can be noted that this problem is non-linear, so it becomes quickly intractable. This is due to the size of the system which grows with the number of components in each subsystem. Thus, making it linear can creates a way to get a solution without fear of the size of the model. The processing method is simple and intuitive (see Coit and Konak, 2006 for details) it is explained in what follows:

Maximizing the problem is equivalent to minimizing a separable function by introducing the logarithm of the original problem and this is due to the fact that the function is monotonic and positive. The transformation steps are as follows:

$$\max_x \left[1 - \prod_{j=1}^{m} (1 - r_{ij})^{x_{ij}}\right] = \min_x \left[\prod_{j=1}^{m1} (1 - r_{ij})^{x_{ij}}\right]$$

The generalization for all the terms of the objective function and by considering the properties of the logarithm function the following expression is obtained:

$$\max_x R(x) = \min_x ln \left\{ \left[\prod_{j=1}^{m1} (1 - r_{1j})^{x_{1j}} \right] \times \left[\prod_{j=1}^{m2} (1 - r_{2j})^{x_{2j}} \right] \times \cdots \times \left[\prod_{j=1}^{ms} (1 - r_{sj})^{x_{sj}} \right] \right\} =$$
$$\min_x \left\{ \left[\sum_{j=1}^{m1} x_{1j} ln(1 - r_{1j}) \right] + \left[\sum_{j=1}^{m2} x_{2j} ln(1 - r_{2j}) \right] + \cdots + \left[\sum_{j=1}^{ms} x_{sj} ln(1 - r_{sj}) \right] \right\}$$

If we suppose that $y_{ij} = -ln(1 - r_{ij})$, the problem of minimization is transformed again to a maximization one, but which can be solved more easily because we get a function composed of separable linear terms as shown in what follow:

$$\max_x \left\{ \left[\sum_{j=1}^{m1} x_{1j} y_{1j} \right] + \left[\sum_{j=1}^{m2} x_{2j} y_{2j} \right] + \cdots + \left[\sum_{j=1}^{ms} x_{sj} y_{sj} \right] \right\} = \max_x \sum_{i=1}^{s} \sum_{j=1}^{m_i} y_{ij} x_{ij}$$

Finally the problem becomes :

$$\max_{x} \sum_{i=1}^{s} \sum_{j=1}^{m_i} y_{ij}\, x_{ij}$$

Subject to :

$$\left|\begin{array}{l} \sum_i \sum_j c_{ij}\, x_{ij} \leq C \\ \sum_i \sum_j w_{ij}\, x_{ij} \leq W \\ \sum_{j=1}^{m_i} y_{ij} x_{ij} > -\ln\left(1 - R_i^{min}\right), i = 1, \cdots, s \\ x_{ij} \in N \end{array}\right.$$

Solution to such problem is now affordable; it suffices to use any optimizing tool such as CPLEX.

1.2.4.2 Binary decision Diagrams

Binary decision diagrams (BDDs) have been discussed earlier. The following example explains how to applied the BDDs to represent the solution relative to the optimization problem. First, we represent the constrains domain, the objective function and both of them.

Suppose a mathematical model composed by a constraint and an objective function are presented as follows:

$$2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$$
$$2x_0 + 3x_1 + 4x_2 + 6x_3$$

The corresponding ROBDD of the constraint and the function are depicted using the graphic representation as shown in figure 1.11.

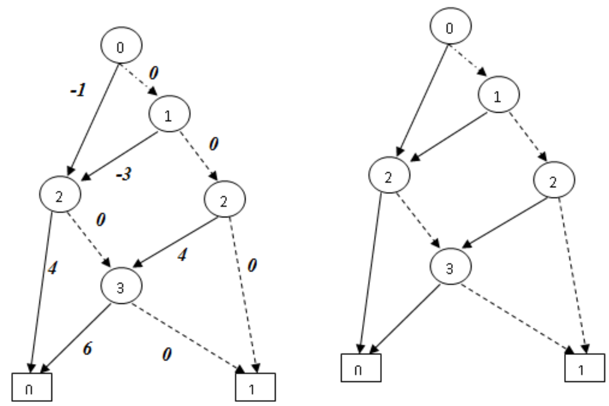


Figure 1.11. Reduced BDD for the constraint $2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$ (left) and for the objective function $2x_0 - 3x_1 + 4x_2 + 6x_3$ (right).

We can add others constraints without any problem and augmenting the dimension of the system of constraints because it suffices to apply the procedure ITE using the logical AND to the BDD structure of the constraints. The conjunction of the BDD's gives a new BDD which represents the system. For example: Suppose that $x_0 + x_1 + x_2 + x_3 \leq 2$ is another constraint. The new BDD built from the conjunction of $2x_0 + 3x_1 + 5x_2 + 5x_3 \geq 7$ and $x_0 + x_1 + x_2 + x_3 \leq 2$ is represented in figure 1.12.

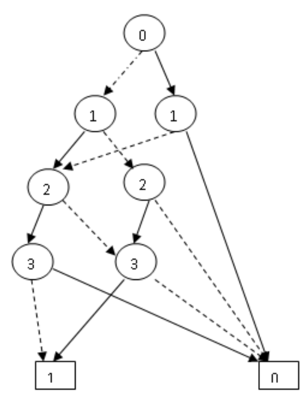


Figure 1.12: Composition of BDD's

Example : The bridge network

Suppose that the corresponding individual reliabilities of each component and the cost are given by the following two vectors:

$R = (0.7, 0.9, 0.8, 0.65, 0.7)$ and $C = (4, 5, 4, 3, 3)$

The intersection of the BDDs generates a new BDD and from which it is shown that the reliability of the system grows from 0,72 if $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 1, 1, 0)$ to 0.99 if $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 0, 0, 3)$.

1.2.4.3 Reliability Importance Measures

The Birnbaum reliability importance measure of a component *i*, denoted by

$$I_B \; i; \mathbf{p} \; = \frac{\partial h(p)}{\partial p_i} = h \;\; 1_i, \mathbf{p} \;\; - h(0_i, \mathbf{p})$$

where $\cdot_i \, , \mathbf{p} \;\; = (p_1, \cdots, p_{i-1}, \cdot, p_{i+1}, \cdots, p_n)$

Thus, the Birnbaum structural importance measure is defined as follows:

$$I_B \;\, i \;\; = I_B \;\; i; (\frac{1}{2}, \cdots, \frac{1}{2}) \;\; = \frac{1}{2^{n-1}} | \;\; 1 \cdot_i \, , x \; : \phi \;\; 0_i, x \;\;\; | > \phi \; \cdot_i \, , x$$

Where | . | denotes the cardinality of a set.

The Birnbaum measure may be interpreted as the probability of the system being in a functioning state with the i^{th} component being the critical one (the failure of this component which is assumed functioning coincides with failure of the system) (Xie and Shen, 1989). It can be remarked that Birnbaum's measure is independent of the i^{th} component.
The following elements will show what type of action to improve the system reliability.

1.2.4.3.1 Definition (from Xie and Shen, 1989)

The Δ-importance of the i^{th} component, $I_\Delta^{(i)}$, is defined as the increase of the system reliability due to an improvement of the i^{th} component. That is

$$I_\Delta^{(i)} = \Delta_i = h \;\; p_i', \mathbf{p} \;\; - h(\mathbf{p})$$

where p_i' is the reliability for the improved component and $h\ p_i', \mathbf{p}$ is the system reliability after the improvement on the i^{th} component.
Using the above equation it can be easily demonstrated that:

$$\Delta_i = (p_i' - p_i).I_B^{(i)}$$

It expected that Δ-importance measure is another equivalent model of Birnbaum important measure.

1.2.4.3.2 Reliable improvement

Another quality of improvement is to fix the new value of the probability of the i^{th} component by a reliable one such as $p_i' = 1$. Thus,

$$\Delta_i = \ 1 - p_i\ .I_B^{\ i}\ = q_i.I_B^{\ i}$$

1.2.4.3.3 Active redundancy

As discussed in section 1, redundancy provision is one effective solution which can improve the reliability of a system. However this type of improvement action can be indentified using the earlier Δ-importance measure and such as the ith component is organized in parallel with another identical. In such case we get :

$$p_i' - p_i = 1 - (1 - p_i)(1 - p_i) - p_i = 2p_i - p_i^2 - p_i = p_i q_i$$

Then the improvement action gives :

$$\Delta_i = p_i q_i.I_B^{\ i}$$

Example :
Call back the undirected bridge network. Let :

$x_1, x_2, x_3, x_4,$ and x_5 are state variables and $p_1 = 0.8, p_2 = 0.3, p_3 = 0.6, p_4 = 0.7,$ and $p_5 = 0.5$.

The function structure is

$$\phi(x) = x_1x_3 + x_2x_4 + x_1x_4x_5 + x_2x_3x_5 - x_1x_2x_3x_4 - x_1x_3x_4x_5 - x_1x_2x_3x_5 - x_1x_2x_4x_5 - x_2x_3x_4x_5 + 2x_1x_2x_3x_4x_5$$

From the earlier equation we can compute what follows:

1- Birnbaum structural importance:

$I_B(1) = 0{,}545;\ I_B(2) = 0.270; I_B(3) = 0.168; I_B(4) = 0.445;\ I_B(5) = 0.250$

2- Δ-importance measure:

$$\Delta_1 = 0.109;\ \Delta_2 = 0.189;\ \Delta_3 = 0.084;\ \Delta_4 = 0.178;\ \Delta_5 = 0.075$$

1.3 State-of-the-Art

Today's, systems are more and more complex, so critical, it appears clear, they must to pursue performing their intended functions without failures over a given period of time. However, as the systems consist of a set of components interconnected by links and subjected to failure, it is more appropriate to use networks for modeling them and one essential aspect of this, is network reliability. Nodes in some cases could also be subjected to failure with a probability value. Networks having both nodes and edges considered as possibly defaulting are simply called imperfect networks. These probabilities are supposed to be statistically independent. Network reliability is a quality which is involved in evaluating the performance of systems.

The primary network reliability consists of evaluating three measures for probabilistic networks. They are:

- *Two*- terminal. Probability that communication is enabled between a source *s* and a destination *t*,
- *K*-terminal. Probability that every node in *K* (a subset of nodes) can communicate with every other node in *K*, and,
- *All*-terminal. Probability that every node can communicate with every other nodes in the network.

Several methods to solve the problem of reliability evaluation have been proposed in the literature. There are those that have been implemented and those who remained in the state of simple theories. There are exact and approximate methods. One major class of exact methods is based on topological methods such as those based on the factoring theorem, the reduction or decomposition (Rosenthal, 1977), (Satyanarayana, 1982), (Wood, 1985). They are more recent and more effective. The reduction operations reduce the size of the network while preserving reliability. They are used to evaluate the reliability of particular networks such as series-parallel graphs in polynomial time (Corinne Lucet, 1993). Factoring methods proceeds by decomposing the network into smaller networks from which the network reliability is deduced by composing small parts reliabilities. For the most complex networks, reduction and decomposition must be combined with the factoring methods to give a very effective tool. A second major class is called enumeration methods. They are two-fold: state enumeration methods and cut/path set enumeration methods. The most basic state based method is complete state enumeration, requiring the generation of 2^m states of a network with m arcs. Cut/path set require enumerating all minimal cut/path set in the graph. These methods involve a substantial research because they are classics in reliability of systems. In these methods, minimal cuts/paths are first determined, and the calculation of reliability is based on them and consists of making the corresponding events disjoint. Three well-known algorithms are used for making them disjoint. They are: Abraham (Abraham, 1979), Heidtmann (Heidtmann, 1989) and recursive-exclusive-exclusion (IE) also known as Poincaré's theorem algorithms (Riordan, 1985). Other similar techniques have been published in many papers, just to name few, they are : GKG ((Veeraraghavan, (1988), (Veeraraghavan, and Trivedi, 1991)), and CAREL (Soh and Rai (1991)). It is simple to note that reliability problems are NP-complete and generating an exact solution is very problematic (Bal, 1986), (Valian, 1979). The desire for fast computation with great accuracy have led to a varied of clever techniques for estimating networks reliability (Colbourn and Harms, 1985). There are two main investigation areas: the estimation of reliability by Monte Carlo sampling techniques, and the bounding of reliability. In the first, simulation consists of generating independent samples and estimating the unknown parameter corresponding to the reliability by an unbiased estimator along with the confidence intervals for the estimate. The relevance of this estimate is related to the number of samples, and their generation. If this number is high, the cost of simulation methods approach or exceed that of the exact methods. In the second, bounding methods attempt to produce absolute upper and lower bounds on the reliability measures from the algebraic structure of the problem. Other intuitive methods use cutsets derivatives instead of pathsets, because in any networks of *m* edges and *n* nodes, the order of the number of cutsets is 2^{n-2} and the order of the number of paths is 2^{m-n+2}, and for a class of networks having nodes of average degree greater than four, as $m > 2n$ and $2^{m-n+2} > 2^{n-2}$ thus such networks have a larger number of paths than cutsets (Rubino, 1988). Example of such networks, fully dense complete network that for n = 10, the number of minimal paths is equal to 109601 and minimal cuts is equal to 256 (Rebaiaia and Ait Kadi, 2011). Also for the 2x100 lattice network, minimal paths number is equal to 2^{99} and minimal cutsets is equal to 10000. Yet, the use of cutsets is much more advantageous than pathsets.

Recently, BDDs as a new formalisms have been developed to minimize the size of networks reliability expressions. They have been overused in different domains, such as electronics and theoretical models in computer sciences called verifiers or simply model-checking. Coudert & Madre (1992) and Rauzy (1993) are the first to introduce BDDs for evaluating networks reliability and since that date a lot of algorithms based on BDDs have been implemented in different tools. Unhappily, realistic tools still away from giving solutions in a reasonable amount of duration-time because the problem has been demonstrated NP-hard (Bal 1980, 1986). So, it still open and can be announced as: “What is the efficient method that can generates a solution for evaluating and thus optimizing the reliability in case of large networks?” The problem is very complex and any method can not by itself give adequate solutions. That is why we propose several algorithms to cover the problematic.

The following objectives can lighten solution we recommend.

1.4 Thesis Objectives

The objectives of the thesis are summarized as follows:

- To classify, test, evaluate and improve techniques and algorithms already published in the literature and those implemented in existing tools.

- To propose, develop and test new approaches and algorithms that are concise, fast and robust for evaluating and optimizing networks reliability.
- To develop techniques for efficient data representation and manipulation during the programming phase.
- To analyse, develop and write programs for testing the proposed algorithms.
- To design a portable tool that supports the functions mentioned earlier and can be also used as a programming platform.

1.5 Literature Review

1.5.1 Minimal Paths set (MPS), Minimal Cuts set (MCS), Sum of Disjoint Products (SDP)

Several algorithms have been developed to enumerate MPS/MCS, most of them require advanced mathematics or can only be applied to either directed or undirected graphs and alternative solutions have been proposed by different authors (Soh and Rai (1993), Jasmon and Kai (1985), Yeh (2009), Al-Ghanim (1999)). Some are specific to the determination of MCS (Patvardhan and Prasad (1996), Lin *et al.* (2003)) and others to MPS (Yeh (2007), Jasmon and Kai (1985)). Some MCS methods are highly related to the MPS so that they are derived from them. Shier and Whited (1985), have proposed a technique for generating the minimal cuts from the minimal paths, or vice-versa. The process is a recursive 2-stage expansion based upon De Morgan's theorems and Quine-type minimization.

Awosope and Akinbulire (1991), present a simple method based on input-reduction programming technique that automates the deduction of MPS/MCS. The method has been applied to a power-system structure in the form of power-arms (termination busbars, branch and protective devices) is the only initial input data needed. The authors argue that the results obtained, in terms of minimal paths and minimal cut-set are similar of those of the literature. In Fotuhi *et al*. (2004), a method called "Path Tracing Algorithm" has been introduced, which can handle both simple and complex networks, and considers both uni-directional and bi-directional branches. As a demonstrating proof the authors explained the procedure using a bridge-network and illustrated by application to a more complicated system. Sandkar et al (1991) propose an algorithm to obtain all path sets that give the required flow at the DC terminal in a power system. By multiplying these path sets, using Boolean algebra, all minimal cut sets that do not transmit the required flow are obtained. From these minimal cut sets, the expression for the probability of failure of transmission of required flow at the DC terminal can be obtained.

Buzacott noted in (Buzacott, 1980) that it is important to well determine the order of the cuts before using the disjoint products version of the cut-based methods. He proposed that the usual approach is to order them in a descending order in terms of the number of arcs in the cut. In Veeraraghavan and Trivedi, 1991, authors describe an efficient Boolean algebraic algorithm to compute the probability of a union of non disjoint sets. The algorithm uses the concept of multiple variable inversions. The paper of Mishra and Chaturvedi (2008) presents an algorithm to enumerate global and 2-terminal cutsets for directed networks. Several benchmark networks have been used to evaluate directed networks with sum-of-disjoint-product (SDP) based multi-variable inversion (MVI) technique without any requirement of complex mathematics or graph-theory concepts.

Author in Yeh (2009) introduces a simple algorithm for finding all MPs before calculating the binary-state network reliability between the source node and the sink node (i.e., one-to-one reliability). It is based on the universal generating function method (UGFM) and a generalized composition operator. The computational complexity of the proposed algorithm is also detailed and an example illustrates generation of all minimal paths. Lin and Donaghey (1993), describe an approach using Monte Carlo simulation to generate minimal path sets by tracing through the system from the input to the output components of the reliability diagram in a random manner. The frequencies' distribution of the minimal cut sets are also determined during the simulation. The paper of Malinowski (2010) presents a new efficient method of enumerating all minimal MPSs connecting selected nodes in a mesh-structured network. This task is fulfilled in two steps. In the first step, the algorithm tries to find all loop-free paths. In the second step, a recursive procedure gradually merges the paths belonging to different paths sets. The authors argue the efficiency of this method by a series of tests. The problem of this method is id due to the backtracking procedure used to deduce all spanning trees. It has shown that it grows exponentially.

The paper (Balan and Traldi, 2003) extends the work of Abraham and Heidtmann by introducing a new preprocessing strategy which works well for SDP algorithms with single-variable inversion (SVI). The authors have observed that optimal preprocessing for SVI-SDP can be different from optimal preprocessing for SDP algorithms which use multiple-variable inversion; one reason for this is that MVI-SDP algorithms handle disjoint minpaths much more effectively than SVI-SDP algorithms do. Both kinds of SDP algorithms profit from prior reduction of elements and of subsystems ---which are in parallel or in series. In Soh and Rai (1991), experimental results are presented showing the number of disjoint products and computer time involved in generating sum of disjoint product (SDP) terms. The authors have considered 19 benchmark networks containing paths (cuts) varying from 4 (4) to 780 (7376). Several SDP techniques are reviewed and are generalized into three propositions to find their inherent merits and demerits. An efficient SDP technique is, then, utilized to run input files of paths/cuts preprocessed using (1) cardinality, (2) lexicographic, and (3) Hamming distance ordering methods and their combinations.

In the study of Yeh (2005), a new algorithm based on some intuitive properties that characterize the structure of MPs, and the relationships between MPs and subpaths are developed to improved SDP techniques. The proposed algorithm is not only easier to understand and implement, but is also better than the existing best-known SDP based algorithm. Based on disjointed algebra and BDD algorithm, Yufang (2010), introduces an improved and simplified algorithm used to solute disjointed MPS. According to the different path length of MPS, he proposes two to disjoint MPSs. He processed for the MP whose length is n-1, keep the original arcs unchanged and add the inversion of those arcs which are not included in the network and get the disjointed result; disjoin the left MP set based on BDD algorithm and realize it through programming. It is shown that the method is efficient and accurate. It provides a new approach for reliability analysis of large scale network system. The application of Binary Decision Diagrams (BDDs) as an efficient approach for the minimization of Disjoint Sums-of-Products (DSOPs) is discussed in Fey and Drechler (2002). The authors tell that the use of BDDs has the advantage of an implicit representation of terms. Due to this scheme the algorithm is faster than techniques working on explicit representations and the application to large circuits that could not be handled so far becomes possible. They showed that the results with respect to the size of the resulting

DSOP are as good or better as those of the other techniques. Locks (1987) describes a minimizing version of the Abraham SDP algorithm, called the Abraham-Locks-Revised (ALR) method, as an improved technique for obtaining a disjoint system-reliability formula. The principal changes are: 1) Boolean minimization and rapid inversion are substituted for time-consuming search operations of the inner loop. 2) Paths and terms are ordered both according to size and alphanumerically. ALR reduces the computing cost and data processing effort required to generate the disjoint system formula compared to the seminal Abraham paper, and obtains a shorter formula than any other known SDP method. Very substantial savings are achieved in processing large paths of complex networks.

More recently, Rebaiaia and Ait-Kadi (2010), propose an elegant and fast algorithm to enumerate MPSs using a modified DFS technique (Tarjan (1972)). The procedure uses each discovered path to generate new MPS from sub-paths. The above procedure is repeated until all MPSs are found. The algorithm didn't at all produce any redundant MPS. More, they extended their work with theoretical proofs and the usage of sophisticated techniques for dynamic data structures manipulation of complex networks.

1.5.2 Reduction and factoring based methods

Factoring algorithm decompositions have been proved to be effective in case the networks are irreducible. It can proceeds on perfect and imperfect networks (Rebaiaia *et al*., 1989), Theologou and Carlier (1991), Simard (1996). They can be applied for directed (Wang and Zhang, 1997) or undirected network, but they have been more worked intensively for the undirected graphs (Satyanarayana and Wood, 1985), (Wood, 1982, 1986), Satynarayana (1980, 1982), Satynarayana and Wood (1985). Moskovitz (1958) is one of the pioneers which has used factoring theorem for undirected networks introduced informally first for minimizing electronic circuits by Moore and Shannon (1956). The principle of factoring theorem is exactly the same as those of Moore and Shannon and the well-known Bayes theorem. All these theorems are a version of the probability total theorem, also known as the conditional probability theorem. Readers are invited to consult books on probability theory. A number of other papers review factoring theorem beginning in the 1970s (Misra ,1970), Murchland (1973), Rosenthal (1974, 1977), Nakazawa (1976). The application of factoring theorem has been cited as a worst case computational complexity and the optimality of classes are NP-hard (Ball (1986), Valian (1977) Chang (1981), Satyanarayana and Chang (1983), and Johnson (1982)). Satyanarayana (1985), proposes a unified formula for analysis of some reliability networks based on Inclusion-Exclusion Poincaré's theorem. The idea is to derive a formula from Poincaré reliability expression that involves the noncancelling terms. He established two theorems for that. In the first he enounces that the domination of a cyclic *K*-graph is always zero and in second, that, the domination of an acyclic *K*-graph with *m* links and *n* vertices is $(-1)^{m-n+1}$. He gives a demonstrating example using 4-nodes bridge network, where he details all the *K*-trees of the network. Satyanarayana and Wood (1985) introduce a new scheme to derive polygon-to chains by reduction on the structure of the network. They proposed seven polygon-to chains reductions. A polygon-to chain reductions is a successive application of factoring theorem on polygons, which must exist as substructures in the graph. Chang (1981) and Satyanarayana and Chang (1983), use a graph invariant $D(G_K)$, called domination, to analyze the complexity of computing *K*-terminal reliability using a factoring algorithm with series and parallel reductions. $D(G_K)$ is equivalent to the number of certain rooted acyclic orientations of G (see Tarjan, 1972). Johnson (1982) discusses other relationships. Satyanarayana (1980) first introduces the concept of minimum domination, $L(G) = \min D(G_K)$ and shows that the lack of a factoring theorem is that directed networks can only be handled in a limited way ((Agrawal, 1974), (Nakazawa, 1976)). Until now, it has discussed the ideal case where network are perfect, in other words, networks subjected to only edges' failure. The problem is so difficult and become more complex if we suppose that also nodes could fail randomly. Such networks are called imperfect networks. Unhappily just few works have been dedicated to the problem. One of the precursors of doing research in this direction is Theologou and Carlier (1991). They have showed using a clever artefact that it is possible to apply factoring theorem with some minor modifications. The problem still opened until a demonstration done by Simard under the supervision of Ait-Kadi (1996) concerning the good way to reduce polygon-to chains in case that both nodes may fail as well as edges. In Rebaiaia *et al*. (2011), a table similar to those of Satyanarayana and Wood (1985) have been established with all the transformation formulas for seven polygons-to chain reductions. Resende (1986) have discusses the design and implementation of PolyChain, a FORTRAN program for reliability evaluation of undirected networks of a special structure via polygon-to-chain reductions. The author presents a small problem and tested it, illustrating the code's output. Theologou (1990) in her dissertation proposes another representation of data structures to optimize the execution time and space memory.

1.5.3 Binary Decision Diagrams Methods

Bryant (1986) was the first to use the work of Akers (Akers, 1978) on the application of binary decision diagrams for symbolic verification of integrated circuits. BDDs have been investigated and implemented first by Bryant (1986, 1992). The problem with BDD representation despite their effectiveness is that, their exponential growing size due to a wrong order declaration between variables. Ruddell (1993) first used an algorithm based on dynamic programming techniques to reduce the size of the BDD and Bollig *et al.* (1996), demonstrate that improving the variable ordering of OBDD is NP-Complete. Coudert and Madre (1992) and Rauzy (1993) applied first, BDDs for evaluating networks reliability. Kuo et al. (1999) used a methodology to evaluate the terminal-pair reliability, based on edge expansion diagrams using OBDD. The algorithm proceeds by traversing the network with diagram-based edge expansion and the reliability is obtained by directly evaluating it on this OBDD recursively. A simple and systematic recursive algorithm has been introduced by *Lin et al.* (2003) that guarantees the generated MCS with ease. This algorithm is combined with OBDD to calculate the reliability of networks. Chang *et al.* (2004) propose an efficient approach based on OBDD to evaluate the reliability of a nonrepairable system and the availability of a repairable system with imperfect fault-coverage mechanisms. Various approaches have been used for the analysis of multi-state systems; examples the BDD-based method (Xing and Dugan, 2002-a), (Tangand and Dugan,2006). (Xing and Dugan, 2002-b). Zang et al. (1999), also proposed efficient approaches based on multi-state BDD (MBDD), and Phased-missions systems analysed using BDD.

1.5.4 Reliability optimization

Many papers have been published to solve the reliability optimization problem and more precisely those addressing the optimal redundancy allocation (Coit *et al.,* Kuo *et al.* (2000), Misra (1991)). Chern shows that the problem of redundancy allocation is NP-hard (Chern, 1992). There are several approaches that provide solutions to such problems. For example, Fyffe *et al*. (1968) use dynamic programming while limiting the problem by considering a single type of component available for each subsystem. To tackle the problem, others models have been used as genetic programming (Coit and Smith, 1996), heuristics (Coit and Konak, 2006)(Coit and Wattanapongsakorn, 2004) and Ant Colony (Kuo *et al.*, 2000). For a useful bibliography the reader is referred to Kuo *et al.* (2000). Recently, Coit *et al.* (2006) used multi-objective programming for the series-parallel systems (figure 1.10) and makes some transformations to translate the problem that is initially non-linear to a linear model whose solution is accessible using CPLEX tool. Also Coit and Konak (2006) present a multi-criteria approach for optimizing the system where the components reliability are estimated with uncertainty. The problem is to maximize the estimated reliability of the system while minimizing the variance associated with it. Another interesting work was published by Ha *et al.* (2006). It solves the problem of optimizing the redundancy by applying heuristics, called as tree-heuristic. This heuristic allows multiple local optimal solutions. Rebaiaia and Ait-Kadi (2010) proposed a solution based on BDDs representation and composition. This approach proceeds first by generating the BDD corresponding to the objective function. A second BDD representing one constraint is associated to the first one. A third BDD is then generated by the composition of earlier BDDs. The algorithm iterates until to cover all the constraints. From the last BDD a solution is generated and the reliability is computed.

One other alternative solution to optimizing the reliability of networks is precisely to act on components which can bring some improvements to the reliability, practically like the redundancy allocation process but the solution is presented differently. It is known that some components are more important than others to the functioning of the system in term of their contribution to the whole system. They are termed measure by component importance or redundancy importance or simply structural importance interchangeable with

Birnbaum's importance index (Birnbaum, 1969). There are other importance indices that have been intensively studied in the literature as Vesel-Fussel importance measure (Meng, 1996) (vasely, 1970). Despite that they are interesting source of information; the problem with such importance indices is they cannot be determined automatically in case the network is complex.

1.6 Outline of the thesis

This thesis considers the problem of modeling, evaluating and optimizing networks reliability. Components and systems are first estimated. Thus, the estimating process is applied in case components failure rate are not available. Two techniques are used based on statistical and historical failure data for the first and the utilization of some guides and norms for the second techniques. Once all data are collected, the network is then drawn in the form of an RBD for Reliability Block Diagrams, which is the basic element on which the algorithms will act.

This thesis includes six contributions presented as articles. Each following chapter represents exactly the content of an article.

In chapter 2, a methodology for modeling and evaluating components and system reliability is introduced. The methodological basis consists of modeling and computing the reliability and the availability of a radio-communication network that meets provincial communication needs. A generic multi-components system is postulated to cover the study followed by some models and rules used to evaluate the reliability of each component, subsystems and the whole system.

Chapter 3 presents an intuitive algorithm to enumerate all MPSs. The algorithm proceeds recursively on the structure of the graph. The presentation treats the problem from both theoretical and practical side. Some theorems have been enounced to cover the algorithm construction which has been implemented and proved sound, complete, simple, compact, modular and easy to be plugged with any software which evaluates the reliability. Experienced using several networks of varied complexities has been tested. The comparison test demonstrated the efficiency of the algorithm in terms of execution time and memory space used by the program.

In chapter 4, a new algorithm is added to enumerate all MCSs of a network. More, it is proved that it is an efficient tool used as input for determining the most relevant components to be augmented for optimizing the reliability of networks.

Chapter 5 clearly shows the efficiency of ROBDD to compute network reliability. For that, an algorithm has been proposed to cover the evaluation of networks reliability. It is a simple data structure representation which can compete with many other solutions. The algorithm takes its input data from the instantaneous paths or cuts generators and gives the value of the reliability.

Chapter 6 proposes a theoretical approach which use factoring theorem to reduce the size of networks, this leads the problem to be of polynomial complexity while it was of exponential complexity. The algorithm proceeds by polygon-to-chain reduction rules and considers the case where network components- vertices and edges could fail randomly.

Chapter 7 presents an extension of chapter 6 to cover other aspects of polygon-to chain reductions. An algorithm is presented and some opportunistic descriptions are described clearly.

The logical link between contributions is showed in the following figure (Figure 1.13.)

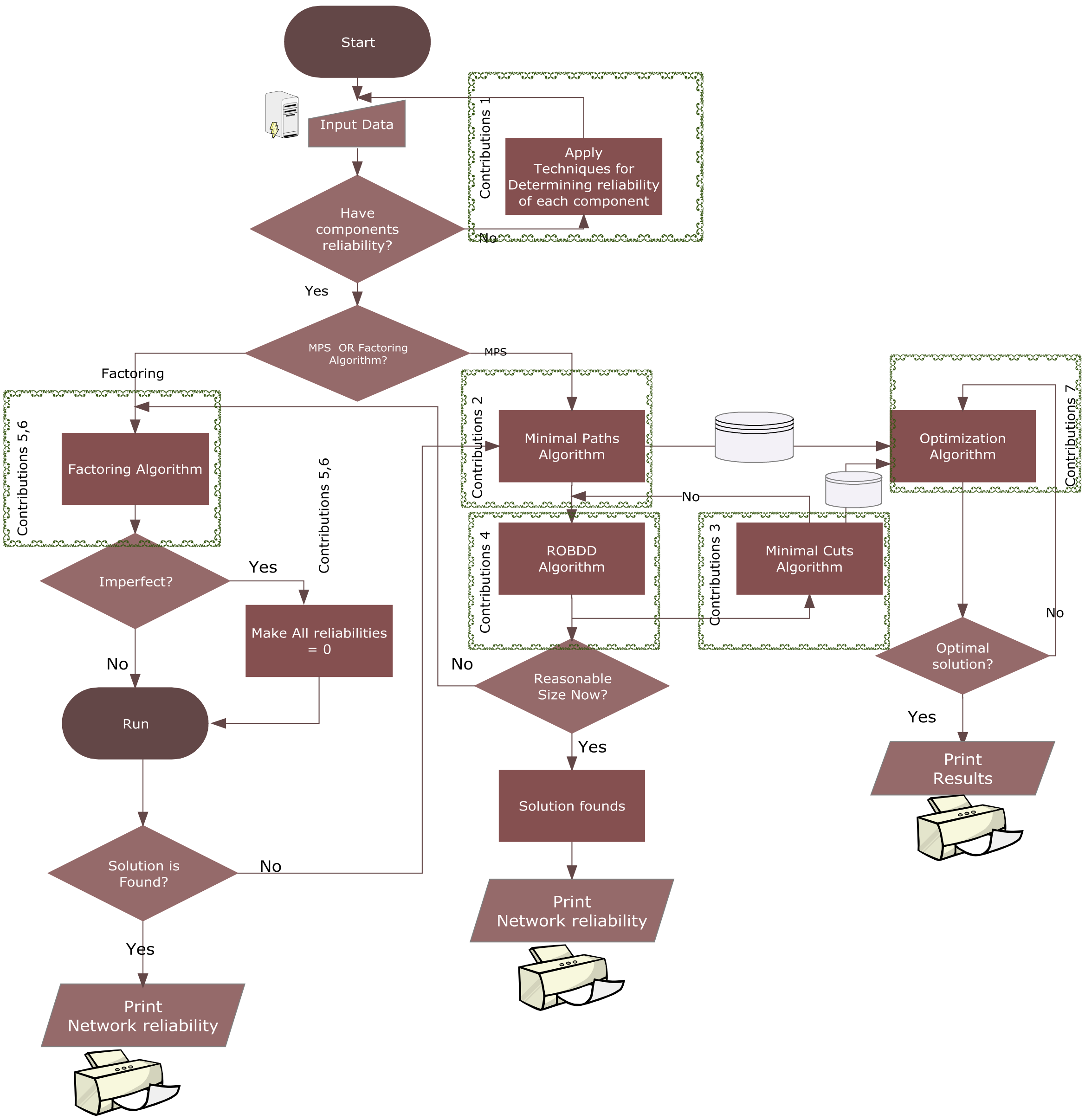


Figure 1.13. Tool Flowchart and thesis contributions

1.7 Conclusion

In this chapter, we introduced the problem of network reliability giving the basic definitions and methods. We presented a concise state-of-the art and a detailed literature review in which it has enumerated the most interesting papers and subjects of network reliability since the first works. Also we presented an outlined which explains the contents of each chapter.

The following chapters contain the results of the thesis research.

References

Abraham J. A., 1979, An Improved Algorithm for Network Reliability, *IEEE Transactions on Reliability*, 58–61.

Agrawal A., 1982, Reliability analysis of rooted directed networks, PhD Thesis, Dept. of IEOR, University of California, Berkeley, 1982.

Akers S. B., 1978, Binary Decision Diagrams, *IEEE Transactions on Computers*, C-27(6), pp. 509–516.

Andersen H. R., 1999, An Introduction to Binary Decision Diagrams, *Lecture Notes, IT University of Copenhagen*.

Ball M. 0., 1986, An overview of the complexity of network reliability computations, *IEEE Transactions on Reliability*, R-36, 1986.

Barlow R. E., Proschan F., 1975, Statistical Theory of Reliability, Holt, Rinehart and Winston.

Birnbaum Z.W, 1969, On the importance of different components in a multicomponent systems, Multivariate analysis II, *Academic Press*, pp. 581-592.

Bollig B., Ingo W., 1996, Improving the Variable Ordering of OBDDs Is NP-Complete, *IEEE Transactions on Computers*, 45(9), pp. 993–1002.

Bryant. R. E., 1986, Graph-Based algorithms for Boolean function manipulation, *IEEE Transactions on Computers*, 35(8), 677–691.

Bryant. R. E., 1992, Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams, *ACM Computing Surveys*, 24(3), pp. 293–318.

Buzacott J. A., 1980, A recursive algorithm for finding reliability measures related to the connection of nodes in a graph, *Networks*, 10, pp. 311-327.

Chang M K., 1981, A graph theoretic appraisal of the complexity of network reliability algorithms, PhD Thesis, Dept. of IEOR, University of California Berkeley.

Chang Y-R, Amari S. V., and Kuo S-Y,2004, Computing System Failure Frequencies and Reliability Importance Measures Using OBDD, *IEEE Transactions on Computers*, 53 (1), pp. 54-68.

Chern, M. S., 1992, On the computational complexity of reliability redundancy allocation in a series system, *Operation Research Letter*, 11, 309-315.

Choi, M. S. and Jun, C. H., 1995, Some variants of polygon-to-chain reductions in evaluating reliability of undirected network, *Microelectronics and Reliability*, 35(1), pp. 1-11.

Coit A. W and Konak A., 2006, Multiple weighted objectives heuristic for the redundancy allocation problem, *IEEE Transactions on Reliability*, 55(3), 551-558.

Coit A. W and Smith A. E., 1996, Reliability optimization for series-parallel systems using a genetic algorithm, *IEEE Transactions on Reliability*, 45(2), 254-260, 1996.

Coit A. W., Jin T. and Wattanapongsakorn, 2004, System optimization considering component reliability estimation uncertainty: a multi-criteria approach, *IEEE Transactions on Reliability*, 53(3), 369-380.

Colbourn C.J and Harms D.D., 1985, Bounding all-terminal reliability in computer networks, Computer Communications Network Group, University of Pengasinan, Technical Report E-123.

Corinne Lucet,1993, Méthode de décomposition pour l'évaluation de la fiabilité des réseaux, Thèse de doctorat, Université de technologie de Compiegne, France.

Coudert O. and J. C. Madre, 1992, Implicit and incremental computation of primes and essential primes of Boolean functions, in *Proc. of the 29th ACM/IEEE Design Automation Conference*, *IEEE, Computer Society Press*, pp. 36–39.

Deo, N. and Medidi, M., 1992, Parallel algorithms for terminal-pair reliability, *IEEE Transactions on Reliability*, 41(2), pp. 201-209.

Dotson, W. P. and Gobien, J. O., 1979, A new analysis technique for probabilistic graphs, *IEEE Transactions Circuits and Systems*, 26, pp. 855-865.

Frank H. and Friech I.T., 1971, Communications, Transmission, and transportation Networks, Addison-Wesley.

Fratta, L. and Montanari, U., 1973, A boolean algebra method for computing the terminal reliability in a communication network, IEEE *Transactions on Circuit Theory*, CT-20, pp 203-211.

Fyffe DE, Hines WW, Lee NK, 1968, System reliability allocation and a computational algorithm, *IEEE Transactions on Reliability*, R-17(2):64-69.

Garey M. R., Johnson D. S., 1979, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co.

Ha C. and Kuo W., 2006, Multi-Path heuristic for redundancy allocation: The tree heuristic, *IEEE Transactions on Reliability*, 5537-43.

Hardy. G., C. Lucet, and N. Limnios., 2007, K-Terminal Network Reliability Measures With Binary Decision Diagrams, *IEEE Transactions on Reliability*, 56, 3.

Heidtmann K.D., 1989, Smaller Sums of Disjoint Products by Subproduct Inversion, *IEEE Transactions on Reliability*, 38, pp. 305-311.

Hopcroft J. E., and Tarjan R. E., 1973, Dividing a graph into triconnected components, *SIAM, J. Computing*, 2, pp 135-158.

Hwang C. L., Tillman F.A and Lee M.H., System reliability evaluation : techniques for Complex/Large systems- a review, *IEEE. Transactions of Reliability*. R-30, pp. 416-423.

Institute of Electrical and Electronics Engineers, 1990, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York, NY,

Jasmon, G. B. & O. S. Kai., 1985, A New Technique in Minimal Path and Cutset Evaluation, *IEEE Transactions on Reliability*, Vol. R-34, no. 2, pp. 136-143.

Johnson R., 1982, "Some Combinatorial Aspects of Network Reliability", PhD. Thesis, Dept. of IEOR, University of California, Berkeley, 1982.

Kuo S-Y, LU S-K andYeh F-M., 1999, Determining pair-reliability on edge expansion diagram using OBDD, *IEEE Transactions on Reliability*, 48 (3), pp. 235-246.

Kuo, W., V. R. Prasad, F.A. Tillman, and C.L. Hwang, 2000, Optimal Reliability Design: Fundamentals and Applications, *Cambridge University Press*, New York, 2000.

Lee. C. Y., 1959, Representation of Switching Circuits by Binary-Decision Programs, *Bell Systems Technical Journal*, 38, pp. 985–999.

Lin, John Y., and Donaghey, Charles E. (1993). Monte Carlo simulation to determine minimal cut sets and system reliability. In *Proceedings of the 1993 annual reliability and maintainability symposium, Atlanta, GA, USA*.

Lin, H. Y., Kuo, S. Y. and Yeh, F. M, 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, in *Proc. of the Eighth IEEE International Symposium on Computers and Communication*, 2, pp. 1341-1346.

Liu, H. H., Yang, W. T. and Liu, C. C, 1993, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronic Reliability*, 33(4), pp. 599-613.

Locks, M. O. and Wilson, J. M, 1992, Note on disjoint products algorithms, *IEEE Transactions on Reliability*, 41(1), pp. 81-84.

Meng F.C., 1996, Comparing the importance of system components by some structural characteristics, *IEEE Transactions on Reliability*, 45(1), pp. 59-65.

Misra K. and Sharma U., 1991, An effective approach for multiple criteria redundancy optimization problem, Microelectronics and reliability, 31(2/3), 303-321.

Misra K. B., 1970, An algorithm for the reliability evaluation of redundant networks, *IEEE Trans. Reliability*, R-19, 1970, pp146-151.

Moore E. F., Shannon C. E.,1956, Reliable circuits using less reliable relays, J. Franklin Institute, 262, pp 191-208.

Moskowitz, Fred, 1958, The analysis of redundancy networks, *AIEE Transactions Commun. Electron.*, vol 39, pp 627-632.

Murchland J., 1973, Calculating the probability that a graph is disconnected, Working Paper No. 182, 1973.

Nakazawa, H., 1976, Bayesian decomposition method for computing reliability of an oriented network, *IEEE Tranactions on Reliability*, R-25, pp 77-80.

Page, L. B. and Perry, J. E., 1988, A Practical Implementation of the factoring theorem for network reliability, *IEEE Transactions on Reliability*, 37(3), pp. 259-267.

Page, L. B. and Perry, J. E., 1989, Reliability of directed network using the factoring theorem, *IEEE Transactions on Reliability*, 38(5), pp. 556-562.

Rebaiaia M-L, Ait-Kadi D., 2011, An efficient Algorithm for Enumerating Minimal PathSets in Networks, submitted to the Journal of Reliability Engineering & System Safety Published by Elsevier.

Rebaiaia M-L, Ait-Kadi D., Merlano A., 2009, “A Practical Algorithm for Network Reliability Evaluation Based on the Factoring Theorem – A case study of a generic radio communication system, *Journal of Quality,* Vol 16, No. 5.

Rebaiaia Md-Li, Ait-Kadi D, Model based Binary Decision Diagrams for Complex Networks Reliability Optimization, IFAC Workshop Programmable Devices and Embedded Systems- PDeS 2010, Poland.

Resende, M. G. C., 1986, A program for reliability evaluation of undirected networks via polygon-to-chain reductions, *IEEE Transactions on Reliability*, 35(1), pp. 24-29.

Riordan J., 1958, An introduction to combinatorial analysis, New York, Wily, 1958.

Rosenthal A., 1974, Computing reliability of complex systems, PhD Thesis, University of California, Berkeley, 1974.

Rosenthal A., 1977, A computer scientist looks at reliability calculations, Reliability and Fault Tree Analysis, eds. R. E. Barlow, J. B. Fussel, N. D. Singpurwalla, *SIAM J.,* pp. 133-152.

Rubino G., 1988, Network reliability evaluation, in State-of-the-art in performance modeling and simulation, K. Bagchi and J. Walrand editors, Gordon and Breach Books.

Rudell, R., 1993, Dynamic variable ordering for ordered binary decision diagrams, in *Proc. of the IEEE/ACM international conference on Computer-aided design*, 42-47.

Satyanarayana A. and Chang M. K.,1983, Network reliability and the factoring theorem, *Networks*, 13, pp. 107-120.

Satyanarayana A., 1980, Minimal domination of graphs, Reliability Seminar Talk, Operations Research Center, University of California, Berkeley.

Satyanarayana A., Prabhakar A., 1980, New topological formula and rapid algorithm for reliability analysis of complex systems, *IEEE Tranactions on Reliability*, R-27, pp. 82-100.

Satyanarayana, A. and Wood R.K., 1985, A linear-Time Algorithm For Computing K-Terminal Reliability In series-Parallel networks, *SIAM J. Comput*., Vol. 14 (4), November, 1985.

Satyanarayana, A., 1982, A unified formula for analysis of some network reliability problems, *IEEE Tranactions on Reliability*, R-31, pp. 23-32.

Shier D. R., D.E. Whited, 1985, Algorithms for generating minimal cutsets by inversion, *IEEE Transactions on Reliability*, R-34, pp. 314- 319.

Simard, C., 1996, Contribution Au Problème D'évaluation De La Fiabilité Des Réseaux, Mémoire de Maîtrise Sciences, *Laval University*, Québec, Canada.

Soh S. and Rai, S., 1991, CAREL: Computer Aided reliability evaluator for distribution computing networks, *IEEE Transactions of Parallel Distribution and Systems*, 2, pp. 199-213.

Tang Z. and J. B. Dugan, 2006, BDD-based reliability analysis of phasedmission systems with multimode failures, *IEEE Tranactions on Reliability*, 55(2), pp. 350–360,.

Tarjan, R, 1972, Depth-first search and linear graph algorithms, *SIAM Journal of Computing*, 1(2), pp. 146-160.

Theologou, O. R. and Carlier, J. G., 1991, Factoring and reduction for networks with imperfect vertices, *IEEE Transactions on Reliability*, 40(2), pp. 210-217.

Theologou, O. R., 1990, Contribution à l'évaluation des réseaux, these de doctorat, and Carlier, Université de Technologie de Compiegne, France.

Tutte W. T., 1966, Connectivity in Graphs, University of Toronto Press, Toronto, 1966.

Valian, L. G., 1979, The Complexity of enumerating and reliability problems, *SIAM Journal of Computing*, 8(3), pp. 410-421.

Vasely W.E, 1970, A time dependent methodology for fault tree evaluation, *Nuclear Engineering Design*, vol. 13, pp. 337-360.

Veeraraghavan M, and Trivedi K.S, 1991, An improved algorithm for symbolic reliability analysis of networks, *IEEE Transactions of Reliability*, R-40, pp 347-358.

Veeraraghavan, M, 1988, Modeling and evaluation of fault-tolerant multiple processor systems, PhD Thesis, Department of Electrical Engineering, Duke University.

Wang W. And Zhang H., 1997, The methods of Reduction in Networks Reliability Computing, *Microelectron Reliability*, 37 (3), pp-461-465.

Wood R. K, 1982, Polygon-to-Chain reductions and extensions for computing K-terminal reliability in an undirected network, ORC 82-12, Operations Research Center, University of California, Berkeley.

Wood R. K, 1986, Factoring Algorithms for Computing K-Terminal Network Reliability, *IEEE Transactions on Reliability*, R-35, NO. 3.

Wood R. K., 1985, "A factoring algorithm using polygon-to-chain reductions for computing K-terminal network reliability", *Networks*, 15, 1985, pp 173-190.

Xie M. and Shen K., 1989, On ranking of system components with respect to different improvement actions, *Microelectrom. Reliab.*, 29(2), pp. 159-164.

Xing L. and Dugan J. B., 2002-1, Analysis of generalized phased-mission systems reliability, performance and sensitivity, *IEEE Transactions Reliability*, vol. 51, no. 2, pp. 199–211.

Xing L. and J. B. Dugan, 2002-2, Generalized imperfect coverage phased-mission analysis, in *Proc. of the Annual Reliability and Maintainability Symposium*, pp. 112–119.

Zang X., H. Sun, and K. S. Trivedi, A BDD-based algorithm for reliability analysis of phased-mission systems, *IEEE Tranactions on Reliability*, 48, no. 1, pp. 50–60.

Chapter 2

A Methodology for Modeling and evaluating the Reliability of a Radio communication Network

2.1 ABSTRACT

Evaluating the performance of a radio communication network needs to check that communications between users are working normally and that hardware/software system components are up and remain ensuring their mission. The methodological basis of this work consists of modeling and computing the reliability and the availability of a radio-communication network that meets the communication needs of an entire province. A generic multi-components system is postulated to cover the study. This chapter presents an exhaustive description of a regional radio communication system followed by some models and rules used to evaluate the reliability of each component, subsystems and the system.

2.2 INTRODUCTION

When defining the performance of complex and critical systems like radio communication networks, it is important to specify system performance across different concepts. Some of these concepts are commonly named dependability, fault-tolerance, reliability, availability, security and survivability. They are defined as indicators from which it is possible to measure the performance of a system. In our study we consider the couple reliability and availability as performance indicators. Note that the IEEE 90 standard (1990) defines the reliability as the ability or the probability of a system to perform its required functions under stated conditions for a specified period of time and availability may be interpreted as the probability that a system is operational at a given point in time. Maintainability is defined as the probability of performing a successful repair action within a given time. Most frequent mathematical expressions of these probabilities are well-known. The reader is invited to consult Ebeling (2005) for precise details.

Nowadays, the analysis of system performance is a crucial test, rough and with major difficulties. This complexity has become indispensable because of the integration of the products of diverse manufactures and technologies (e.g. Motorola, Cisco). Computing the reliability of a system depends of the specification of the reliability of each piece of equipment being powered. Missing one piece will generate false values in computing the system performance and can have dramatic consequences. For instance, a manufacturer of equipment may specify the failure rate estimated using an accelerated degrading testing when no failures statistics are available.

In this chapter, we present some of the results recorded in the report of a research project realized by our team as consultants. The project has been entrusted by the telecommunication services of the Quebec government (Canada). The main objective was to assess the reliability of the radio communication network covering the entire province of Quebec. This requirement condition was an essential request of public and private users for the renewal of their service contracts. As the system is a new network that has just been installed in several stages since early 2000, therefore, the first finding is to be sure that the network as such, does not suffer from breakage of certain materials which have not had a disastrous consequence on the functioning of the network. That said, at first sight, it is assumed that the system is correct and one could roughly estimate the reliability of the network on the basis of this assumption. Thus, this information is not a concrete proof to be given to customers such that to testify that the network is highly reliable without giving them an exact value such as 95%. For this purpose, our assessment made sense of quantifying the reliability of the network as a value, the most precise possible.

Following our experience in the conduct of the project, it has shown that network failures can occur for a variety of reasons. In most cases, the causes are accidental, inexperienced technicians, natural (wind, frost, ice) and shortage of spare parts. A preliminary description has been published in Rebaiaia *et al* (2009) and Ait-Kadi *et al.* (2009).

The approach presented herein is comparable with the study done by Willett *et al.* (1988). The difference between the two works is that, their methodology is applied to a shortwave broadcast relay station and is evaluated in term of the number of broadcast hours delivered to the assigned coverage areas. The authors present three models: one for covering the reliability, another for the availability and a third one for the maintenability. However the availability was a culminating parameter. Our work is broader and considers a network consisting of 156 receiving and transmitting stations in addition to a modern technology using digital hardware and software components connected using TCP/IP communication protocol links. Contrary to their objective our project considers reliability as being essential to the definition of the performance of the network.

Other interesting work have influenced our study for conceptualising the network as separate subsystems working together Moubray (1977) and Noll, M (www.columbia.edu). The separation supposes that each subsystem is itself a network and thus the radio communication network becomes a network of networks. Although, this decomposition increase security and efficiency but provides a simplified calculation of the reliability and the availability and consequently the maintenability becomes intuitive.

Many others research helped in doing the project as an other side of the work for developing tools for maintaining the network using standard and guidelines as those described by Moubray (1977), Agrawal *et al.* (1996) and Michelin *et al.* (1988). These insights have been of great help.

The present document also describes some characteristic of a digital radio communicating system which uses dynamic assignment of frequency (ADF). The system offers a set of roaming functions. It allows any user to communicate with his group or another group, whatever the area provided it is covered in a limited space where he can be detected by receiver's sites.

The lack of data that can determine the reliability or availability of a product requires the us e of models and standards for determining the failure rate or for checking the Mean Time B etween Failures (MTBF) value under an exponential distribution. This study also concerns t he design of a model to describe and evaluate the reliability of a radio telecommunication s ystem using standard for assessing the reliability prediction of electronic equipments and co ntains formulae for modeling and computing the failure rate of a component under some co nditions as in Ait-Kadi *et al.* (2009).

The organization of this chapter is as follow. In the next section readers are briefed on the architecture of the radio communication network. We give an approximate description of the system and some useful definitions. Section 3, presents the main lines to model the components of the system and calculate their reliabilities. Finally concluding remarks are given in section 4.

2.3 ARCHITECTURE OF THE RADIO NETWORK

This section presents the architecture of the radio communication system. It details the most important components and introduces their characteristics which can be used to determine the reliability. First of all, we introduce a methodology to describe the system element by element and computes the reliability of each of these products and finally it help to build the RBD for reliability block diagram to determine the reliability of each subsystem and thus the reliability of the system. In the following we present the notion of generic system instead of system in any way.

2.3.1 Definition of radio communication system

A conceptual design for a generic radio system provides a convenient mean for people to communicate instantaneously engaged in various public safety-related services. The generic radio system is usually expected to transmit and receive to its coverage areas radio signals that are used to carry voice and data on a daily basis throughout the year and whatever the atmospheric conditions. An example of a radio communication system is presented in Figure 2.1. This figure gives a simplified description of the network. We can see that each node of the network represents a radio communication station called site. There are standard sites and master sites. They are described below. Any two sites communicate using a system of microwave data transport.

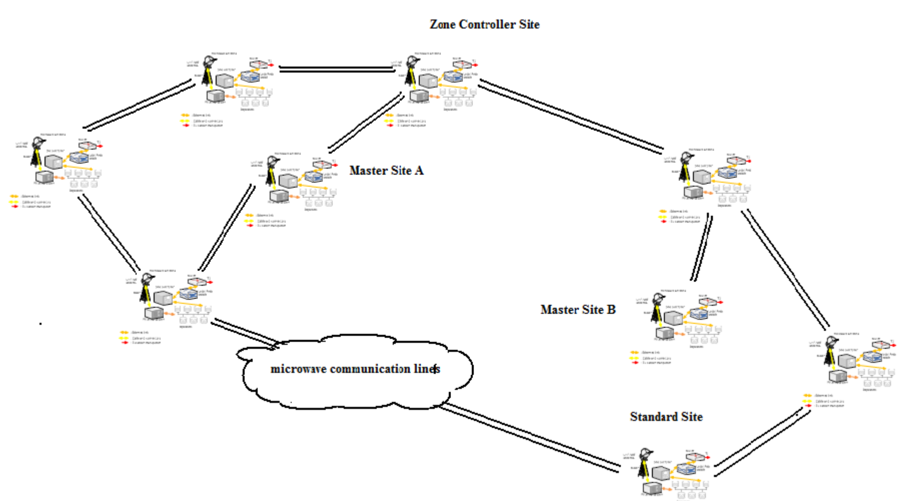


Figure 2.1. Radio communication network

The network is composed by the following necessary elements:

- *Radios* (portable and mobile).
- *Sites* (master sites, secondary Radio Frequency (UHF/VHF sites)).
- *Zones* (in a zone or zones there is one or more UHF/VHF sites).
- *System* (single zone or multiple zones with one or more UHF/VHF sites)

In the radio system, a zone is responsible for managing its own elements (sites, repeaters, subscribers, UHF-VHF and microwave carriers) interconnected using a high-speed transport network to form a Wide Area Network (WAN). This WAN system is composed by hardware/software devices and routers allowing users configuration information, call processing information, and audio to be conveyed throughout the system. Each zone framework includes a physical infrastructure, managing mobility and processing calls transported using IP (Internet Protocol) packet technology through the network.

2.3.2 Standard and master sites

A basic radio system consists of equipment for transmitting and receiving radio signals that are used to carry voice (audio) or data.

The radio sites is equipped with one or two antennas for broadband coverage on which is terminated 4 to 8 transmitter-receiver transponders (Tx/Rx). The transponders are connected to each antenna via filtration equipment of type Multicoupler. The multicouplers form a chain of multicoupling able to accept others transponders in expansion (see Figure 2.2, and Figure 2,5.). Figure 2.2, gives an overview of a *standard radio* site which consists of the following subsystems:

1- Outdoor subsystem

- A tower
- UHF-VHF antennas
- Microwave antennas
- Connectors
- Power electric energy Resources deserved by a public network.

2- Indoor subsystem

- Cable RG-393

- Switch
- Inside connector

3- Batteries used to secure power alimentation in case the power network fails to deliver electric energy.

4- Communicating control system which is the heart of communication coordination between user's solicitations, radio control manager and call processing mobility systems. Figure 3 shows the most important devices and their line connections. You can see that the system is fully redundant. There are site controllers; Ethernet switches, remote routers and a repeater which is an RF station that serves as the RF link between the system and the mobile and portable units. In a site there are a minimum of two and a maximum of 28 repeaters which have each one of them, (1) a receiver to pick up the RF signal from the subscribers, (2) a transmitter to send RF signals to the subscribers and (3) a wireline interface to send audio to a centrally located device used for dispatching functions.

A master site consists of core and exit routers, WAN and LAN switches, controllers and some operative computers plus others monitoring and dispatching hardware/software systems such as core, exit and gateway routers, AEB, PBX, dispatching consoles Elite and others.
For more precision the reader is invited to see Ait-Kadi *et al.* (2009). Note that the main objective being to develop an analysis methodology for Reliability, Availability and Maintainability (RAM) and thus to optimize the performance of the system.

Based on the RAM requirement it is expected to determine:

- Reliability of the critical components and subsystems.
- Overall reliability of the radio-communication system.
- Availability of the critical components and subsystem.
- The sensitivity of the RAM parameters.
- The maintenance time and budget for corrective actions.

- The maintenance time and budget for preventive operation as a function of subsystem.

The approach used in this work is based on the well-known standard STD-MIL 217-F (1995), which specifies that reliability prediction utilizes a series model for system reliability evaluation for one or more components. It is permitted in case of critical components or according to the structure of a subsystem to use hot-standby and the redundancy items are modelled in parallel.

The reliability of the overall system is critically dependent upon the time duration each component is operating. The MTBF of each component is used if it is available otherwise failure statistical data are used for determining the failure rate. The reliability of each component for a time duration T can be expressed as an exponential function depending on the relative MTBF. Table 2.1 to table 2.6 give the expected reliability, the Mean Time Between Failures and the failure rate.

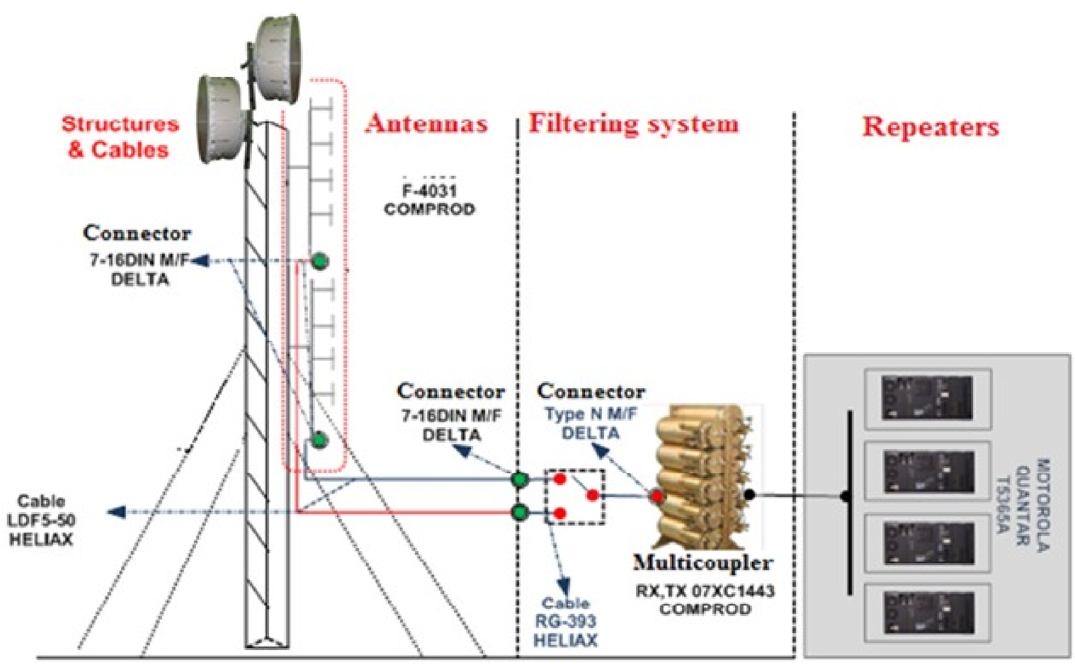


Figure 2.2. A standard radio communication station

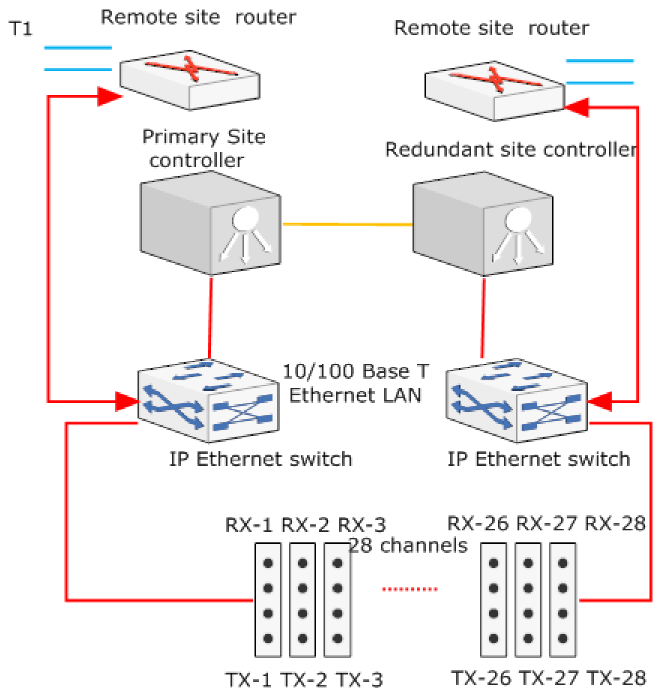


Figure 2.3. Communicating system inside a radio system.

2.4 MODELLING OF A RADIO COMMUNICATION SYSTEM

Information about the electric system was provided using service of 43 between 134 stations (32%) over a period of approximately of two years. The reliability for the battery system was calculated using data from a fixed proportion of a sample of 990 cells of 78 operational batteries in 71 stations (53%), for a period approximately equal to two years and a half. There was no data on the preventive and corrective maintenance, no more records of electrical and mechanical failures of some elements like the DC-AC converter and the AC Distribution Panel. For that, we developed an electric model to approximate the AC Distribution Panel and DC-AC converter, based on the predictive model MIL-HDBK-217F-2. Also. it was supposed that the failure rate of each electronic or electric element is constant according to the norm MIL-HDBK-217 F. In what follows we will show how to model the different components and especially those who cannot be determined by statistical methods. As all the components are electric or electronic types, using an exponential probability distribution would be better representative for the failure rate approximation.

2.4.1 Modeling the antennas

The antennas have been developed on a specific request with redundant features and sound quality according to the following characteristics:

- Anodizing Type II according to MIL-A-8625F,- Standard ISO 9002, W47.2FM1987 (c1998),
- CAN/CSA-S37-01,
- Wind Zone D (600p),
- Zone III glass (40mm),
- Reliability class 1 (factor of importance 1),

and respecting the following standards:

- MIL-C-39010
- MIL-C-83446
- MIL-HDBK-217F Notice 2 (11.2)

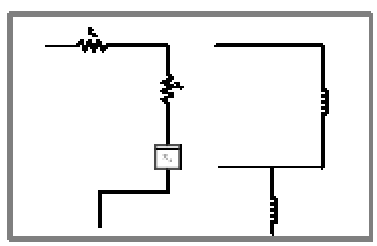


Figure 2.4. Electric model for the antennas

Antenna failure rate is derived from the following model:

$$\lambda_p = \lambda_b \times \pi_T \times \pi_Q \times \pi_E$$

Where :

$\lambda_b = 0.00005$ (basic failure rate corresponding to the variable coil)

$\pi_T = 1.4$ (Temperature: THP $50C^0$-60^0)

$\pi_{Q} = 2$ (Quality: Between military and standard qualities)

$\pi_{E} = 12$ (Environmental factor)

$\lambda_p = 0.00005 \times 1.4 \times 2 \times 12 = 0.00168$ (failure/10^6 hours)

The corresponding reliability using exponential distribution is reported in the following table:

Table 2.1. Antenna reliability value

Reliability	**0.99999999832**
Failure rate	0.00000000168
MTBF(h)	595238095.2381
FITS	1.68

2.4.2 Modeling the Outdoor Connectors

The parameters identified are :

λp = Failure rate

λ_b = Basic failure rate

π_k = Degradation factor due to plugging and unplugging connectors

π_p = Active pines factor

π_E = Environnemental factor

The expressions of the outdoor connectors failure rate is :

$\lambda p = \lambda_b \times \pi_K \times \pi_P \times \pi_E$

λp =0.0040 (material C $30C^0 - 40C^0$) × 1.5 (0.5 – 5 in 1000 hours) × 1.4 (two active elements) × 14 (GM elements subjects to random movements)

$\lambda p = 0.1176$ (failures/10^6 hours)

Table 2.2. Outdoor connectors' reliability value

Reliability	**0.99999988240001**
Failure rate	0.0000001176
MTBF(h)	8503401.3605442
FITS	117.6

2.4.3 Modeling the Filters (Multicouplers)

The multicouplers are used to connect multiple receivers or radio antennas. They consist of coaxial cavities resonating at ¼ wavelength loop coupling input/output and coaxial cables whose lengths can be calculated based on the used frequencies. They isolate each Tx / Rx (Transmission-reception) on channel to reduce the interference problems (noise transmitter and inter modulation disturbance) on the radio site.

As the technical data are not available, the only obvious solution is to use guidelines contained in the standards outlined in what follows :

- MIL-C-92
- MIL-R-12934
- MIL-C-3607-83517
- MIL-HDBK-217F Notice 2(9.1-10.1-15.1)

The schematic representative of the filtering system is given in Figure 2.5, and the electric model simulating its features is shown in Figure 2.6.

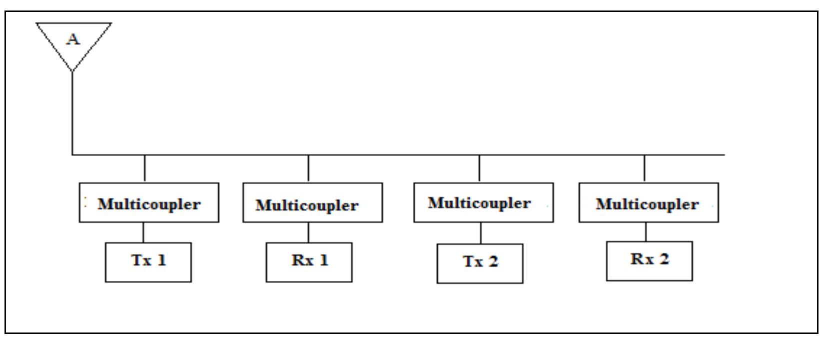


Figure 2.5. Filtering system

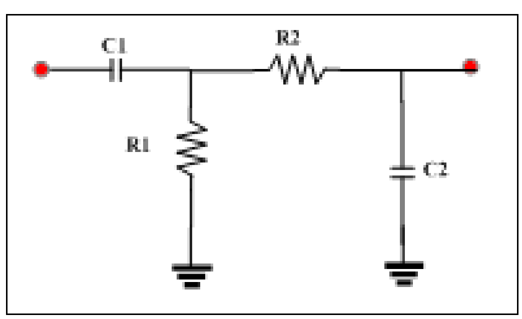


Figure 2.6. Electric model to simulate filtering system

The following model gives the failure rate of each multicoupler where (CT) is the parameters of the capacitor and (RT) is those of the resistance. The coupling of capacitor and resistor generate the reliability of the multicoupler devices. The failure rate expression is as follows:

λp = Failure rate of the capacitor
λ_b = Basic failure rate
π_T = Temperature factor
π_C = Capacitor factor
π_V = Voltage stress factor
π_{SR} = Series resistance factor
π_Q = Quality factor
π_E = Environmental factor

So to calculate the failure rate of a capacitor we use what follows :

$\lambda p = \lambda_b \times \pi_T \times \pi_C \times \pi_V \times \pi_{SR} \times \pi_Q \times \pi_E$

$\lambda p = 0.0000072 \times 2 \times 1 \times 5 \times 1 \times 1.25 \times 1$
$\lambda p = 0.000009$ (Failure/10^6hours)

The reliability of the capacitor is :

$R_x = e^{-0.00000000009} = 0.99999999991$

The parameters of the Resistance

$\lambda p = \lambda_b \times \pi_T \times \pi p \times \pi_S \times \pi_Q \times \pi_E$
$\lambda p = 0.0024$ (variable Resistance) $\times 2 \times 1 \times 1.1$
(table MIL-HDBK 217F9.1) $\times$ 1 (between military and Standard quality) $\times$ 1 (easy control of the maintenance)

$\lambda p = 0.0024 \times 2 \times 1 \times 1.1 \times 1 \times 1$

The failure rate of the resistance is :

$\lambda p = 0.00528$ (failure / 10^6 hours).

And the reliability of the resistance is :

Ry = 0.99999999472

As the capacitor and the resistance operate in series, their reliabilities composition is :

$R_{CR} = R_x \times R_y = 0.999999999917 \times 0.99999999472$

$R_{CR} = 0.999999994637$

The following table resumes the values of the reliability, the failure rate, the mean time between failure and the FITS (= 10^9 Failure rate).

Table 2.3. Filtering system reliability

Reliability	**0.999999884637**
Failure rate	0.00000000536300001438
MTBF(h)	186462800.17127
FITS	5.36300001438

2.4.4 Modeling the indoor connectors

Using the same model as for the outdoor connectors, we find the value of the parameters as in the following table:

Table 2.4. Indoor connectors' reliability

Reliability	**0.99999977587203**
Failure rate	0.0000000126
MTBF(h)	79365079.365079
FITS	12.6

2.4.5 Modeling the Radios (QUANTAR)

The MTBF of the radio *Quanta* has been provided by the manufacturer. Thus, if we use the value of the MTBF we can deduce the failure rate and the reliability as follows:

MTBF = 110000 hours

$$\lambda_p = \frac{1}{110000} = 0{,}0000025974025 \text{ (Failures hour)}$$

Table 2.5. Quantar reliability value

Reliability	**0.99999740260078**
Failure rate	0.0000025974025
MTBF(h)	385000
FITS	2597.4025974026

2.4.6 Modeling the power electric station

To find the reliability we have used the management information system to retrieve faults information of each radio communication station. This is another way by which to find the failure rate relative to each electric power station. Figure 2.6 presents information representing fault warnings and their durations. They are given by the followings statistics:

- Sample of 43 stations over a period of 11,520 hours
- Total Number of failures 512 hours
- Total Idle Time: 937 hours 12 minutes and 11 seconds
- Average length of inactivity by default: 1 hours, 49 minutes and 50 seconds
- Average length of inactivity per station: 21 hours, 47 minutes and 44 seconds.

From these data, we determine the value of the reliability as depicted in the following table:

Table 2.6. Electric station reliability

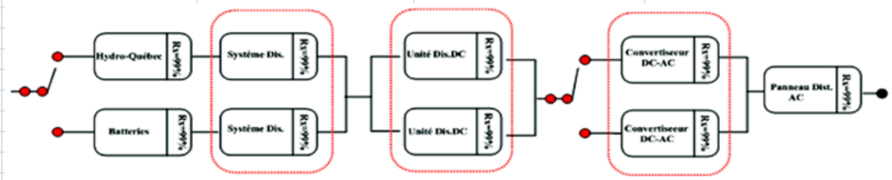
Reliability	0.95652874
Failure rate	0.04444444
MTBF(h)	22.50
FITS	4444444

At the end, we evaluate the reliability of the global electric installation system as detailed in the figure 2.7.

Note that figures 2.8, 2.9 and figure 2.10 give the reliability of other parts of the system.

Element	MTBF(h)	FITS	Failure rate	Reliability
Power	22,50	44444443,5067082	0,0444444435067082	0,956528740000001
Batteries	4461736,15	224,128	0,0000002241280	0,999999775872025
Dist. System	385000,00	2597,40259740260	0,00000259740259740260	0,999997402600776
Uni.Dist.DC	256100,00	3904,72471690746	0,00000390472471690746	0,999996095282907
Panneau Dist. AC	100000000,00	10,0	0,00000001	0,999999990000000
Converter DC-AC	35511363,14	28,16000039649	0,00000002816000039649	0,999999971840000

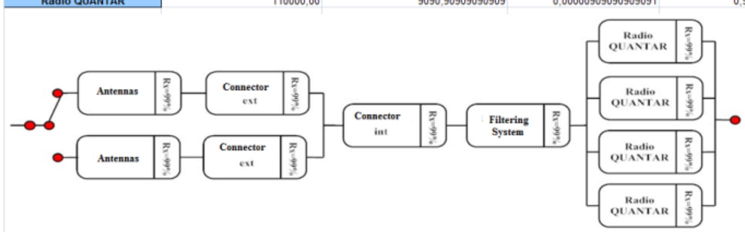
(ideal switch)
0,9998000



Hydro-Batterias (std-by)	0,999996123560677		
Sys.Distri. (Parallel)	0,999999999993254		
Uni.Dist.DC (Parallel)	0,999999999984753		
Converter DC-AC (std-by)	1,0	Reliability	0,999996113538722
		MTBF(h)	257302,9769912000
		FITS	3886,4688302235
		Taux Panne	0,0000038864688302235

Figure 2.7. Reliability evaluation of the electric power system

Element	MTBF(h)	FITS	Failure rate	Reliability
Antennnas	595238095,00	1,68000000067	0,00000000168000000067	0,999999998320000
Connector(ext)	8503401,36	117,600	0,0000001176000	0,999999882400007
Filtres (Multicoupler)	186462800,17	5,36300001438	0,00000000536300001438	0,999999994637000
Connector(int)	79365079,00	12,60000005796	0,00000001260000005796	0,999999987400000
Radio QUANTAR	110000,00	9090,90909090909	0,00000909090909091	0,999990909132231



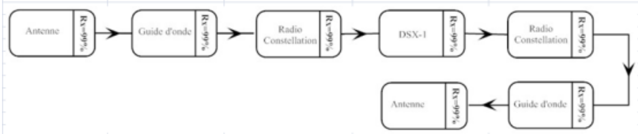
Antenna -Connector (standby)	0,999999999999993	UHF-VHF System Reliability	0,9999999820369890
Radio Motorola(p) 2 parmi 4	0,999999999999996		
		MTBF(h)	55669955,1955016
		FITS	17,9630106848
		Taux Panne(h)	0,0000000179630106848

System Overview	
General	
Mean Availability (All Events):	0.99903411
Std Deviation (Mean Availability):	0.024984676
ean Availability (w/o PM & Inspection):	0.99903411
oint Availability (All Events) at 131400:	0.9981
Reliability(131400):	0.9981
Expected Number of Failures:	0.0019
Std Deviation (Number of Failures):	0.043549738
MTTFF:	69091095.78
ystem Uptime/Downtime	
Uptime:	131273.082
CM Downtime:	0
Inspection Downtime:	0
PM Downtime:	0
Total Downtime:	126.9180135
ystem Downing Events	
Number of Failures:	0.0019
Number of CMs:	0
Number of Inspections:	0
Number of PMs:	0
Total Events:	0.0019
Costs	
Total Costs:	0
Throughput	
Total Throughput:	0

Point Results at Preselected System				
Time	A(t)	R(t)	(1-A(t))%	(1-R(t))%
13140	0.9999	0.9999	0.01	0.01
26280	0.9997	0.9997	0.03	0.03
39420	0.9994	0.9994	0.06	0.06
52560	0.9991	0.9991	0.09	0.09
65700	0.9989	0.9989	0.11	0.11
78840	0.9989	0.9989	0.11	0.11
91980	0.9988	0.9988	0.12	0.12
105120	0.9984	0.9984	0.16	0.16
118260	0.9982	0.9982	0.18	0.18
131400	0.9981	0.9981	0.19	0.19

Figure 2.8. Reliability evaluation of the RF communication system

Element	MTBF(h)	FITS	Failure rate	Reliability
Antennas	75000	13333,3333333	0,0000133333333333	0,999986666755555
Waveguide	150000	6666,667	0,0000066666667	0,999993333355556
Radio Constellation	28034,9590749004	35669,7506612484	0,0000356697506612484	0,999964330885497
DSX-1	28921188,857	34,57672521501	0,00000003457672521501	0,999999965423275



System Overview	
General	
Mean Availability (All Events):	0.605437727
Std Deviation (Mean Availability):	0.361952611
Mean Availability (w/o PM & Inspection):	0.605437727
Point Availability (All Events) at 10000:	0.3343
Reliability(10000):	0.3343
Expected Number of Failures:	0.6657
Std Deviation (Number of Failures):	0.471768764
MTTFF:	9094.753291
System Uptime/Downtime	
Uptime:	6054.377266
CM Downtime:	0
Inspection Downtime:	0
PM Downtime:	0
Total Downtime:	3945.622734
System Downing Events	
Number of Failures:	0.6657
Number of CMs:	0
Number of Inspections:	0
Number of PMs:	0
Total Events:	0.6657
Costs	
Total Costs:	0
Throughput	
Total Throughput:	0

Total Reliability	0,9998886321238150
MTBF(h)	8978,7499706275
FITS	111374,0780477610
Failure rate	0,0001113740780477610

Point Results at Preselected System Times						
Time	A(t)	R(t)	(1-A(t))%	(1-R(t))%	Cost(t)	Mean A(t)
1000	0.8965	0.8965	10.35	10.35	0	0.946436381
2000	0.7987	0.7987	20.13	20.13	0	0.896621383
3000	0.713	0.713	28.7	28.7	0	0.849192541
4000	0.6422	0.6422	35.78	35.78	0	0.805948613
5000	0.5744	0.5744	42.56	42.56	0	0.766342701
6000	0.5136	0.5136	48.64	48.64	0	0.729269527
7000	0.4609	0.4609	53.91	53.91	0	0.694690696
8000	0.4187	0.4187	58.13	58.13	0	0.662821352
9000	0.3765	0.3765	62.35	62.35	0	0.633310721
10000	0.3343	0.3343	66.57	66.57	0	0.605437727

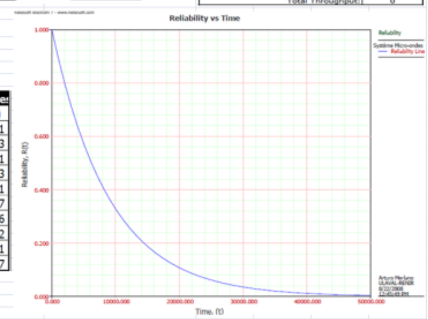
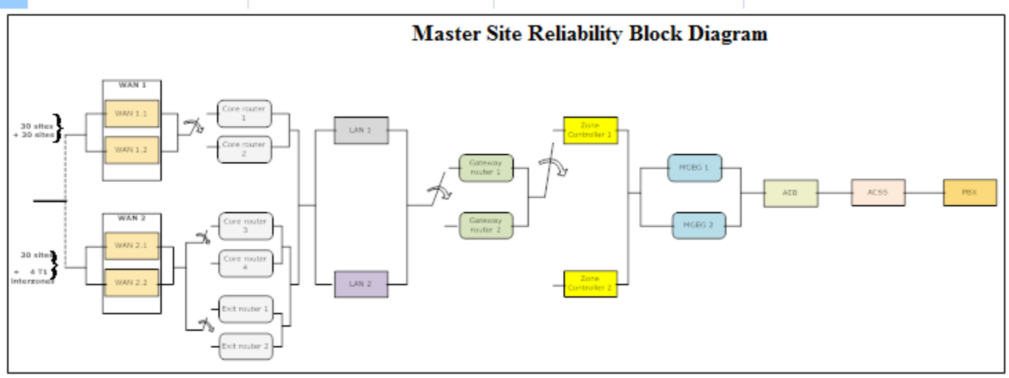


Figure 2.9. Reliability evaluation of the microwave transmission/reception system

Element	MTBF(h)	FITS	Failure rate	Reliability
WAN 1	2773837,00	360,51145038443	0,00000036051145038443	0,999999639488615
WAN 2	2773837,00	360,51145038443	0,00000036051145038443	0,999999639488615
Core Router	100000,00	10000,00000000000	0,0000100000000000000	0,999990000050000
Exit Router	100000,00	10000,00000000000	0,0000100000000000000	0,999990000050000
LAN	70983,00	14087,88019666680	0,00001408788019666680	0,999985912219037
Gateway	100000,00	10000,00000000000	0,0000100000000000000	0,999990000050000
Zone Controlle	567461,00	1762,23564262566	0,00000176223564262566	0,999998237765910
MGEG	100000,00	10000,00000000000	0,0000100000000000000	0,999990000050000
AEB	350000,00	2857,14285714286	0,00000285714285714286	0,999997142861224
ACSS	200000,00	5000,00000000000	0,0000050000000000000	0,999995000012500
PBX	196000,00	5102,04081632653	0,00000510204081632653	0,999994897972199



MGEGpar	0,999999999900001
ZC	0,999999999998447
GatewaySBY	0,999999999950000
Fiabilité CoreRouterSBY	0,999999999950000
Fiabilité ExitRouterSBY	0,999999999950000
WanPar	0,999999999999870
LanPar	0,999999999801534
SerieWan1CoreRouter	0,999999999949870
SerieWan2CoreRouter3sby	0,999999999999870
Reliability router	1,000000000000000
ParalWANcores	1,000000000000000

Total Reliability	0,999987040550284
MTBF(h)	1,0000129596
FITS	999987040,5502840000
Failure rate (xh)	0,0000129595336904152

Fig 2.10. Reliability evaluation of a Master site system

2.5 CONCLUSION

This chapter describes a methodology for developing a reliability model which is applicable to evaluating the performance of a radio-communication system. A generic multi-components system is postulated to cover our study. We present in this document some parts of the system and we give details for just a few ones. We note that in most cases, a lack of statistics disrupt the fact to obtain highly representative parameters. Despite this, it is possible at each time to find a method to compute the reliability of a system; it suffices to use MIL-HDBK-217 F. Next soon, we planned to compute the reliability of the global system using the model developed here.

REFERENCES

Rebaiaia, M. L.,2009, Ait-Kadi, D., and Merlano, A., A practical algorithm for network reliability evaluation based on the factoring theorem- A case study of a generic radio communication system, *Journal of Quality*, vol.16, N0. 5.

Ait-Kadi, D., Rebaiaia, M. L., and Merlano, A., 2009, Évaluation et optimisation du réseau national intégré de radio communication de la province du Québec, Canada, Rapport de travail et d'expertise.

MIL-HDBK-781A, 1996, Reliability test methods, plans, and environments for engineering, development, qualification, and production, *MIL-HDBK-781A*, US DOD, 1996, pp. 32-42.

MIL-HDBK-217 F, 1995, Reliability Prediction of electronic equipment, *MIL-HDBK-217* F, US DOD.

Michlin, Y.H., 2008. Improvement on Sequential Testing in MIL-HDBK-781 A and IEC 61124, *IEEE Transactions on Reliability*, Vol. 57, N0. 2, 379-387.

Agrawal A. K. and Holzman E. L., 1996, Active Phased Array Architectures for Reliability, in *Proc. of the IEEE International Symposium on Phased Array Systems and Technology*, p. 159-162, Print ISBN: 0-7803-3232-6.

Willett J. H., Harvey D. L, and Hagn G. H., 1988. A methodology for modeling the reliability availability and maintenability (RAM) of a generic shortwave broadcast relay station, *IEEE Transaction on Broadcasting*, vol. 34, N0. 2.

Noll, M., Network Security and Reliability (Emergencies in Decentralized Networks), www.columbia.edu/dlc/wp/citi/citi517.html

Moubray J, 1977, Reliability-Centred Maintenance II, 2nd ed. Industrial Press, New York., Institute of Electrical and Electronics Engineers. I

EEE Standard Computer Dictionary, 1990, A Compilation of IEEE Standard Computer Glossaries. New York, NY.

Ebeling C. R., 2005, An introduction to reliability and maintenability engineering, *Waveland Press, inc, Long Grove Illinois*.

Chapter 3

An efficient Algorithm for Enumerating Minimal Paths Set in Networks

3.1 ABSTRACT

Operation of a system can be modelled by a network whose components can be represented by nodes and their functional relationships by arcs. The reliability of complex networks is a very sensitive issue which requires implementing powerful methods for its evaluation. Many algorithms have been proposed to solve networks reliability problem. Among them minimal pathsets and minimal cutsets are techniques which have been used intensively last decades and implemented in different tools. The enumeration of minimal pathsets can be obtained for example by a simple application of the well-known Diskjtra algorithm. The problem of enumerating minimal paths and cuts set is not easy for large systems, so efficient techniques are needed because the generating process time grows with the dimensions of the network. This paper presents an efficient algorithm to enumerate all the minimal paths of a network. The algorithm proceeds recursively using a depth first search procedure. The major theoretical bases have been demonstrated sound and complete and the program is simple, compact, modular and easy to be plugged with any software which evaluates the reliability. It has been implemented and experienced using several networks of varied complexities. The comparison test demonstrated the efficiency of the algorithm in terms of execution time and memory space occupied by the program.

ACRONYM

MPS/MCS	Minimal pathsets/cusets.
DFS	Depth first search algorithm.

NOTATION

$G(X, E)$	Network with node set X and the edge set E, where **t** and **s** are respectively the source node and the sink node. G may be directed or undirected.
$n = X$,	Number of nodes of the network.
$m = \|E\|$	Number of nodes of the network.
variable	a variable represents an edge or a node.
e_{ij}	$e_{ij} \in E$ is a directed edge from node i to node j.
p_{e_i}	Functioning probability of e_i

p_{ij}	Functioning probability of the edge taken between node i and node j

3.2 INTRODUCTION

The concept of MPS/MCS is a very effective tool to determine the reliability of a system from the disjoint form of its terms. A path *P* is defined as a set of adjacent nodes connected using edges (network components) so that if all the components are failure-free, the path is considered as up and leads the network to be up. A path *P* is minimal if it has no proper subpaths. Conversely, a cut *C* is a set of edges such that their removal leads the network to fail. A cuts *C* set is minimal if no proper cut exists. Technically, this means that the failure of the cutset components ensures that the entire system fails and the failure of a minimal path component discard the expression path from computing network reliability. Enumerating all the MCS may be a feasible way to evaluate the reliability if the number of MPS is too huge to be enumerated practically. One example of this kind of networks is the complete network (10 x 10) which contains 45 edges and 109601 MPS and 2x100 lattice networks contains 2^{99} MPS and 10000 MCS (Lin *et al.* (2003)). Several algorithms have been developed to enumerate MPS/MCS, most of them require advanced mathematics or can only be applied to either directed or undirected graphs and alternative solutions have been proposed by different authors (Soh and Rai (1993), Jasmon and Kai (1985), Yeh (2009), Al-Ghanim (199)). Some are specific to the determination of MCS (Patvardhan and Prasad (1996), Lin *et al.* (2003)) and others to MPS (Yeh (2007), Jasmon and Kai (1985)). Some MCS methods are highly related to the MPS so that they are derived from them. Shier and Whited (1985), have proposed a technique for generating the minimal cuts from the minimal paths, or vice versa. The process is a recursive 2-stage expansion based upon De Morgan's theorems and Quine-type minimization. Jasmon and Kai (1885) used an algorithm which proceeds in two steps. The first step concerns the deduction of link cutsets from node cutsets and the second deduce the basic minimal paths using network decomposition. So, in addition to the enumeration of cutsets directly, it is possible to obtain them from the inversion of minimal paths (Shier and Whited (1995)). Al-Ghanim (1999) presents a heuristic programming algorithm to generate all MPS. The algorithm proceeds by creating a path, then iterates back from an explored node in the current path using unexplored nodes until to reach the source node. The procedure uses each discovered path to generate new MPS from subpaths. The above procedure is repeated until all MP are found. The problem with Al-Ghanim's algorithm that it produces redundant MPS, which needs a tool to avoid them using extensive comparisons. Recently, Yeh (2009) presents a simple algorithm for enumerating all MPS between source and the sink nodes. It is based on the universal generating function. Before that, Yeh (2007) proposes a simple heuristic algorithm for generating all minimal paths. The algorithm proceeds by adding a path, or an edge into a network repeatedly until the network is equal to the original network. Also, Liu *et al.* in (1993) and Shier and Whited (1985) have used two different algorithms showing that a 2×100 lattice network has 2^{99} paths but contains 10,000 minimal cutsets.

3.3 NETWORKS MODELING

Let $G(X,E)$ be an arbitrary graph in term of direction and let P be an arbitrary family of (source-sink) pairs of nodes of $G: P = \{(s_i, t_i) \in E \times E \mid i \in [k] = \{1,\ldots,k\}\}$. It is assumed that $s_i \neq t_i$ for all $i \in [k]$. Consider that each edge and each node could be subjected to random *s*-independent failure occurrences and are weighted with a probability p_i. For a specified set $K \subseteq X$ of G, we denote the K-terminal reliability of G by $R(G_K)$. When the cardinality of K is 2 $(|K| = 2)$, it is called 2-terminal (or terminal-pair) reliability which defines the probability of connecting the source node with a target node. In the most cases, it suffices to have a 2-terminal relation to evaluate the reliability of networks. The generalisation of the problem is called the K-terminal reliability, and considers the subset K $(|K| > 2)$ differently from the 2-terminal reliability. A success set, is a minimal set of the edges of G such that the nodes in K are connected; the set is minimal so that deletion of any edges causes the nodes in K to be disconnected. Topologically, a success set is a minimal tree of G covering all nodes in K. In the same way, by using the conjunction of all of minimal paths we can evaluate the reliability of the network. There is another way to compute the reliability by considering minimal cuts. They can be derived by inverting the terms of minimal paths or by determining them using algorithms or heuristics (Abraham (1979), Heitdtmann (1989)).

Some definitions are necessary to introduce the problem of MPS/MCS. Let that:

- Each node and each edge have two states: working or failed. The states of edges are s-independents.
- The graph is connected and free-loops.
- In case of parallel edges, they are systematically replaced by one edge whose reliability is obtained using parallel relation, see (Rebaiaia *et al*. (2009)) for more details.

$$p_{i,j} = p_i + p_j - p_i p_j$$

- If two edges are in series, the second is deleted and the reliability of the first one is replaced by the product of the two edges reliabilities.

$$p_{i,j} = p_i p_j$$

In the following we introduce preliminaries, definitions, lemmas and theorems. They are presented as follows:

3.3.1 Definitions

1- A path of length q is a chain $u = (x_1, \dots, x_q)$ in which the terminal endpoint of arc μ_i is the initial endpoint of arc μ_{i+1} for all $i < q$. Hence, we often write $\mu = (x_1, x_{k+1})$, where k is the number of edges.

2- The dimension of a path $P = (x_1, -, x_{k+1})$(we note *dim(P)*) is equal to the number of edges which is k. Nodes x_1 is called the initial endpoint and node x_{k+1} is called the terminal endpoint.

3- A graph G is said to be *connected* if between any two nodes $x, y \in X$ there exists a chain $\mu = (x, y)$.

4- A graph G is said to be *quasi-strongly connected*, if for all $x, y \in X$, there exists a path $\mu = (x, y)$ or a path $\mu = (x, y)$

5- A path P in a graph G is said to be a *1*-path if any two nodes $x, y \in X$ of the path, they are linked by only one edge. Any path in a graph G which is a *1*-path is said to be a branch.

6- A 1-path P is said to be maximal, if and only, if it connects the n nodes of the graph. Thus, the dimension is $\dim(P) = |X| - 1 = n - 1$.

7- An arborescence is defined as a tree that has a root. In other words see *Corollary 1*.

8- Node x_i is a brother of node $x_j\,(i \neq j)$ if node x_i and node x_j have the same parent.

9- A partial graph of $G\ X, E$ is the graph $G\ X, V$ whose node set is X and edge set is V such that the graph G without the edges $E - V$.

10- Let $e_r = (x, y) \in E$, a co-edge is another edge $e_m = (x, y) \in E$. In other words, $e_r || e_m$, they are parallel and form a cycle.

11- PathSet (PS): PS is the set of 1-paths in a graph G connecting the source s and the sink t.

The following Lemma explains how to evolve in a branch.

Lemma 3.1. At any node in the generation tree T, if T, a) contains a vertex v that is marked (has been visited), and b) does not have an edge to any children node in X that it is not marked, then the branch need not be expanded further and it is needed to backtrack to proceed the research from another *brother* node if it exist.

The proof of this Lemma is trivial because the above paragraph explains the procedure.

Corollary 3.1. ((Berge (1973)) page 35). A graph G has a partial graph that is an arborescence if, and only if, G is quasi-strongly connected.

Property 3.1. A graph G with at least two nodes and an edge, the deletion of the edge separates s from t, and thus discard this link from computing the reliability.

Theorem 3.2. (Let H be a graph of order n > 1.) The following properties are equivalent (and each characterizes the arborescence):

(1) H is a quasi-strongly connected graph and this property is destroyed if we add to *H*, an arc of *G* not included in *H*, thus we create a cycle.

(2) If each arc of *G* not included in H can be added to H and creates a cycle, this arc is a cocycle and the set of such arcs constitutes a basis of independent cocycles of dimension *n-1*. Thus we can create $m - n + 1$ cycles.

(3) H is quasi-strongly connected and if we can create $m - n + 1$ cycles, thus we generate $m - n + 1$ different minimal paths.

Let us demonstrate just a part of this theorem so that (1) => (2) and (2) => (3).

Proof:

(1)=> (2), is simple to be demonstrated, it follows theorem 13 ((Berge (1973), page 30)) and the above definition 4. From property (1), H is quasi-strongly connected graph and thus it is connected and without cycles. Thus H is a tree. Therefore H has $m - n + 1$ arcs.

(2)=> (3). If we can create one cycle by connecting any two node of H using an edge of *X* not included in *H*, then we can travel through *H* using such edge belonging to a new path. Thus if the set of possible created cycles is equal to $m - n + 1$ then we create $m - n + 1$ different paths.

Theorem 3.3. In a (directed or undirected) graph with *n* nodes, if there is a 1-path from node x_i to node x_j $i \neq j$, then the dimension of such path is maximal and equal to *n-1*, thus the path is also minimal and the graph contains only one MPS.

Proof: if we suppose that the 1-path uses all the nodes of the graph, such that no node occurs more than once, it is normal that two nodes are used as terminal ends of an edge and for linking three nodes it is needed a sequence of two edges. Recursively we deduce that *n-1* edges are necessary for linking *n* nodes. Thus the path is maximal. Also, suppose that the path is of dimension *n-1* and it is not minimal then if we add another edge to the path, then necessary such edge will link two nodes of the path which are already linked by an edge. Thus we create a cycle and the path loses its properties. We conclude that the path is minimal.

Theorem 3.4. In a (directed or undirected) graph with n nodes, if there is at least one minimal path (MPS) from node s to node t $s \neq t$, then the dimension of such path is maximal and equal to dim $MPS = q$ where $q \leq$ n-1 edges. In other words, the maximal length of an MPS is equal to n-1.

Proof : The demonstration of theorem 4 is a consequence of theorem 3, and thus it can be deduced easily because if we suppose that the path is of dim $MPS = q$ then it uses $q - 1$ edges and if we add another edge to such path, either we create a cycle and thus the MPS forget its properties or we get another path of dimension $q + 1$, which is not minimal because we have linked the terminal node with another node of the graph.

Theorem 3.5. Given a graph $G(X, E)$. Suppose that G contains a path such that $|X| = n$. The following properties are equivalents:

(1) Path MPS is a branch of G of dimension q, and by adding a new edge $e_r \in E$ to the branch we create a new cycle. Necessary the edge e_r is a co-edge of the existing one.

(2) If such cycle exist and is of dimension 2, the deletion of the corresponding co-edge of e_r creates a new MPS,

(3) The set of all the MPS constitutes an arborescence whose node root is s and the leaves are all t.

Proof: (1)=>(2). Path MPS is a 1-path and thus it is a branch by definition 7. Its dimension is equal to q by theorem 4, so by adding any edge $e_r \in E$ in $E - (edges\ of\ MPS)$ so that to link any two successive nodes we create a co-edge and thus a cycle (not a circuit in case of directed graph). So, if we delete the edge e_r from the cycle we create a new MPS exactly the same of the first one but with a new edge replacing e_r. Thus we have created a new MPS from s to t.

(2)=>(3). If the deletion of each co-edge create an MPS and thus a branch, thus all the branches constitutes the ramifications of a graph tree and thus an arborescence with s as root and the occurrence of t are the terminal nodes of the branches which are the leaves.

(3)=>(1). As each ramification from root s to leave t constitutes a path and by the properties of arborescence, each path is a minimal path and thus a branch.

The following theorems assure that MPSs are unique and the set of MPSs is complete.

Theorem 3.6. The algorithm generates the MPSs without repetition.

Theorem 3.7. The algorithm does not miss any MPS.

Theorem 3.8. All the generated MPSs form a basis of independent paths, so that each element of the vector corresponds to an edge. If the graph is undirected the sign of the vector element is not represented and if the graph is directed, the sign must be positive.

Thus, the demonstration of theorem 3.6 and theorem 3.7 is simple due to the theorems 3.1 to 3.5 and *lemmas 1.* This is explained by the fact that the algorithm constructs the 1-path step by step without missing any edge not marked. For theorem 3.8, the MPSs are independent because each MPS is different from the immediate past-generated one by at least one edge component, and this is recursively unrolled on the successive MPSs, which constitutes a basis of independent vectors.

3.4 A PROCEDURE FOR MINIMAL PATHSETS ENUMERATION

To enumerate MPSs it is important to look over nodes and edges composing a path from the source until reaching the sink nodes. There are different ways of doing that. Several algorithms, heuristics and metaheuristics have been proposed to determine MPS. One of the recent works is due to Yeh (Yeh, (2009)). It is based on a simple Universal Generating Function method to search for all MPS in a Network. The algorithm involves simple recursive procedure combined with simplification. Another good algorithm is presented by Colbourn (1987). The problem of evaluation the network reliability is an NP-hard problem and enumerating MPS is also NP-hard (Ball (1980), Ball (1986)).

In the following we present a fast procedure for deducing minimal pathsets. The kernel of the procedure uses a recursive function based on the depth first search algorithm (Tarjan (1972)).

Note : The complexity of the depth first search algorithm is $O(|V| + |E|)$.

For more precision, minimal paths are generated directly from the graph structure by traversing the graph using depth first search algorithm. To perform the research, the algorithm uses dynamic data structures for memorizing the intermediary values of three stacks. Each execution cycle corresponds to a minimal path. This algorithm is called recursively as many times as there is a possibility to reach the terminal node by a new branch.

Suppose that G is the graph model and *s* and *t* are respectively initial and terminal nodes. S is a stack data structure used to memorize successive edges forming a minimal path. Note that stack S works dependently of two others stacks (they are not appeared in the following algorithm). A stack S1 is used to mark the explored nodes during the research in each branch of the graph. The marking of S1 is a function to avoid the redundancy and assures the condition so that each path is minimal. Stack S2, contains the MPS and another stack called S3, is used to memorise the position of an edge in the successive list of edges. This helps to trace the minimal path. For looking for the MPSs in graph branches, the algorithm unrolls a tree which has its root in the first edge of the MPS. .

The algorithm proceeds recursively and always begins using the initial node. The procedure explores one of the adjacent edges and continues until the terminal node is found so that one MPS has been discovered. Then by backtracking action the last node is used to get onto another branch.

The following pseudo-algorithm gives an overview of the procedure.

It is clear that a recursive call is present in the body of the main program and inside the procedure PathDFS. The calls assure the fact to find a minimal path and then to go back to try to find another one. The algorithm stops when the operation of backtracking and forwarding didn't find any edge not marked.

3.5 ALGORITHM

The algorithm has been programmed in the last versions of MatLab and Java. Note that each state is illustrated using the well-known Bridge Network. Table 1, details the execution steps of the algorithm and Figure 3.2, shows the minimal pathsets generated by the algorithm.

Note that, the stack S is a generic data structure which replaces the stack S1, S2 and S3.

The following pseudo-Algorithm formally presents the MPS generator and illustrates it with two examples. The first one with precision and the second gives all-MPS. The program generates each MPS exactly once, without missing any MPS, and without duplication.

```
Algorithm stack S = pathDFS(G, v, z)
   setLabel(v, VISITED)
   S.push(v)
   if  v = z
     return S.elements()
   for all e in G.incidentEdges(v)
     if getLabel(e) = UNVISITED
          w ← opposite(v,e)
     if getLabel(w) = UNVISITED
         S.push(e)
         pathDFS(G, w, z)
         S.pop(e)
     else
      S.pop(v)
end

Program Main()
     Input: A connected graph with node set, edge set, a  source node, and a sink
node .

  Declaring dynamics vectors and stacks (put in them zeros)
  Declaring initial and terminal nodes (v, z)
   Do While .true.
     pathDFS(G, v, z)
  if "the last minimal path have been encountered"
     return .false.
   enddo
Output: All MPS in the graph.
```

3.6 Detailed Description of the Algorithm

Based on the discussions presented in section II and the present section in accordance of the above pseudo-algorithm and theorem 8 which insures the construction of the MPSs, such that no duplicate MPS is generated and all the MPSs are minimal, we propose the following heuristic algorithm. The heuristic gives more details than those presented in the above pseudo-algorithm.

1. Assign node numbers sequentially from 1 (source node) to N the sink node (e.g. bridge network: node 1: 1, node 2: 2, node 3: 3 and node 4: 4).
2. Create automatically the adjacency matrix representation of the network (Figure 3.1 (b)), and a dynamic empty pathsets matrix so that its dimension is null.
3. Start from the initial node 1 (mark it using stack S1) and generate the next nodes call them Next (Next = children(node 1)). Note that Next is a dynamic vector and its size is the number of children. At the beginning it is a null vector.
4. Check if the encountered node is different from the sink node. If so, check if all the elements of S1 have been marked. If so, go to 8, otherwise go to 5.
5. For each element of the vector check if the node has been marked, if it is so, go to 6, otherwise mark it and put the edge (node parent-node children) in the stack S2 and mark the position of the relative edge in the stack S3 and go to 7.
6. If the element has been marked, backtrack to return to another brother node. If it exists go to 5, otherwise go to 4.
7. Go forwarding until the sink node has been encountered. If so, copy S2 in the pathsets matrix, backtrack and go to 5.
8. Print the pathsets matrix and Stop.

Note that the application of the algorithm using Bridge network (Figure 3.1) is illustrated in Table 3.1 as explained in the following illustration.

3.6.1 Illustration Step-by-Step Example

Consider a 4-node, 5-edges bridge network with its adjacent matrix enumerated by the order of edges taken from 1 to 5 (Figure 3.1 (b)) (*step 2*). Note that we have numbered 1 the source node and 4 the sink node (*step 1*).

Step 3. Mark the node 1, so the S1(1) = 1, go to step 4.

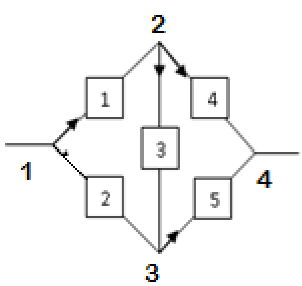
Step 4. Node 1 $\neq$ Node 4 go to step 5.

Step 5. Determine children(node 1): children(1) = {2,3}. Node 2 is the first node and it is not marked because S1(2) $\neq$ 1. Then, we mark Node 2, and S2(1) = 1 (edge between node 1 and node 2) and the position of edge 1 in the stack S3 is marked (S3(1) = 1). Go to step 7.

Step 6 and Step 7. We determine the children of node 2 etc. We continue alternating steps 4, 5 and 6 until the sink node is encountered. If so backtrack (see figure 3.1 (a), then (b) and (c)).
Step 8. Print MPS matrix and STOP.

The results of the above illustration are detailed in table 3.1. We can see that at the beginning, the stacks are empty and their dimensions are null. They receive at each step of the algorithm certain value numbered by 1 if a new node is added to the path. A position in stack S2 corresponds to the encountered edge where all the edges of a path are represented. Stack S3 is an indicator of the edge position. When a MPS is built and the terminal node is compared, the algorithm decrements the last position of the stack and continues to do so until a non marked edge is found. Then a new MPS is generated and followed by a third one.

3.6.2 Example 3.1:



(a)

Nodes	1	2	3	4
1	0	1	2	0
2	0	0	3	4
3	0	0	0	5
4	0	0	0	0

(b)

Figure 3.1. Bridge network (a) and its relative adjacency matrix

Table 3.1. Minimal paths set enumeration of the bridge network

Operations	Stack S1	Stack S2	Stack S3
	Nodes i	Edges j	Edge position

	s_1 s_2 s_3 s_4	x_1 x_2 x_3 x_4 x_5	P_1 P_2 P_3 P_4 P_5
Top(push(i,s)) = i	1	x_1	0
Top(push(i,s)) = i	1 1	x_1 x_3	1 - 1 - -
Top(push(i,s)) = i	1 1 1	x_1 x_3 x_5	1 - 1 - 1
Remove(i, s) = s	1 1	x_1 x_3 -	1 - 1 - -
Remove(i, s) = s	1	x_1 -	1 - - - -
Top(push(i,s)) = i	1 1	x_1 x_4	1 - - 1 -
Remove(i, s) = s	1	x_1 -	1 - - - -
Remove(i, s) = s	-	-	- - - - -
Top(push(i,s)) = i	1	x_2	- 1 - - -
Top(push(i,s)) = i	1 - 1		
Remove(i, s) = s		x_2 x_5	- 1 - - 1
Remove(i, s) = s	1 - 1	x_2 -	- 1 - - -
Isempty(s) = true Stop.		-	- - - - -

The minimal pathsets are :

MPS = { $\{x_1, x_3, x_5\}$; $\{x_1, x_4\}$; $\{x_2\ x_5\}$ }.

The structure function of the network is:

$$\varphi(x) = 1 - (1 - x_1x_4)(1 - x_2x_5)(1 - x_1x_3x_5)$$

which corresponds to a new representation of the network where the MPSs are considered in parallel. Thus, the reliability of the network is equal to the mathematical expectation of the structure function $E(\varphi(x))$. It is computed after using Boolean simplification rules:

$$R(G) = E(\varphi(x)) = r_1r_4 + r_2r_5 + r_1r_3r_5 - r_1r_2r_3r_5 - r_1r_2r_4r_5 - r_1r_3r_4r_5 + r_1r_2r_3r_4r_5.$$

and if $\forall r_i = 0.9$ (the individual reliabilities corresponding to each edge), then $R(G) = 0.97119$.

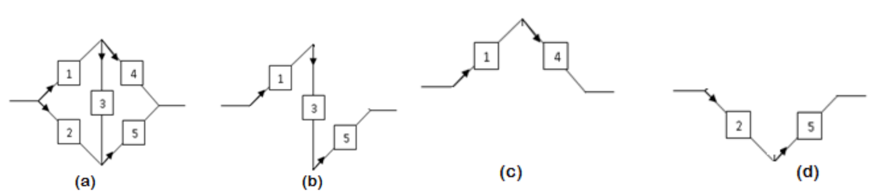


Figure 3.2. (a) The bridge network. (b) the 1st first MPS. (c) the 2nd MPS and (d) the 3rd MPS.

3.6.3. Example 3.2:

a. Directed graph

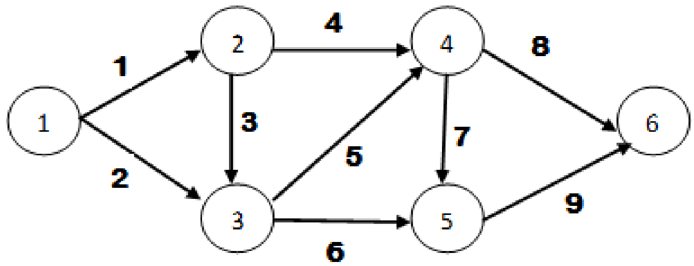


Figure 3.3. A 6-node, 9-link example network (directed)

#	Minimal Pathset
1	1, 3, 5, 7, 9
2	1, 3, 5, 8
3	1, 3, 6, 9
4	1, 4, 7, 9
5	1, 4, 8
6	2, 5, 7, 9
7	2, 5, 8
8	2, 6, 9

b. Undirected graph

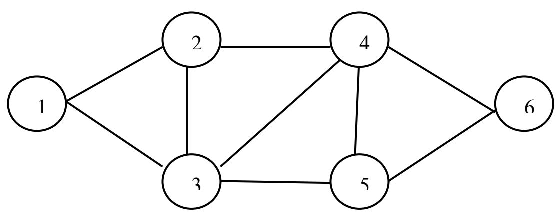


Figure 3.4. A 6-node, 9-link example network (undirected)

Table 3.2. Adjacent matrix of the network in Figure 3.4.

G =

nodes	1	2	3	4	5	6
1		1	2			
2	3		4	5		
3	6	7		8	9	

4		10	11		12	13
5			14	15		16
6				17	18	

All the MPSs generated from the network in Figure 3.4 are:

#	Minimal Pathset
1	1, 4, 8, 12, 16
2	1, 4, 8, 13
3	1, 4, 9, 15, 13
4	1, 4, 9, 16
5	1, 5, 11, 9, 16
6	1, 5, 12, 16
7	1, 5, 13
8	2, 7, 5, 12, 16
9	2, 7, 5, 13
10	2, 8, 12, 16
11	2, 8, 13
12	2, 9, 15,13
13	2, 9, 16

3.7 EXPERIMENTAL EXAMPLES

The examples are all taken from the literature. We have compiled examples which have been used as benchmarks for demonstrating the implementation of our algorithm. The run times shown in the examples are determined by the execution of the algorithm. For our results the host computer was a simple laptop personal microcomputer and the implementation was compiled by Java 1.6.1 and MatLab 2009.

The times required for the execution of the algorithm using the benchmark of the figure 3.5 are presented in table 3.3. This table gives the number of MPS and the corresponding execution time. We can notice that the time is in seconds and so it is the smallest amount by comparison with those found in the literature. We have used some others networks with density variants. The results are shown in figure 6 and figure 7. Figure 9 represents the execution time per one path for a complete graph of dimension 10 nodes with a variation of density equal to [0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. 1]. We can notice that despite the number of MPS which grows exponentially, the time per one path falls to a small value. Note that all the times values are in seconds/1000 and are the effective execution times and do not include the time required to perform I/O operations. Our program solves the 2-terminal minimal pathset enumeration problem and thus it can be used as input procedure for evaluating networks reliability. The program can also be extended to be used for the case of K-terminal networks. So the implementation uses dynamic data structures such that each effective representation is performed by dynamic lists and queues structures. So at each time of the execution processing, the system keeps only one cell to represent physically any element used to memorize or to compute a part of the calculus. However, it is preferable to use a computer system with a processor and a memory manager that runs under 64-bit.

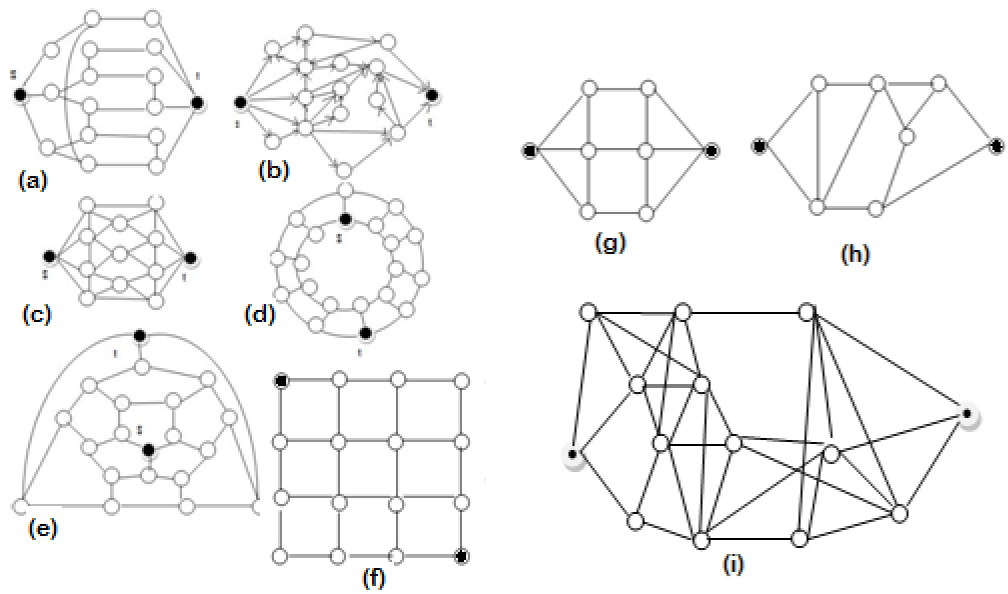


Figure 3.5. Benchmark Networks

Table 3.3. Detailed experimental results.

Graph	# Nodes	# Edges/Arcs	# MPS	Time (sec)
a	17	23/46	136	0.106287
b	16	30	36	0.043412
c	13	30/60	3972	1.577472
d	20	29/58	432	0.274192
e	20	29/58	516	0,265083

f	16	24/48	184	0.209079
g	8	13/26	29	0,033207
h	8	12/24	24	0.021653
i	14	36/72	42036	248.21824

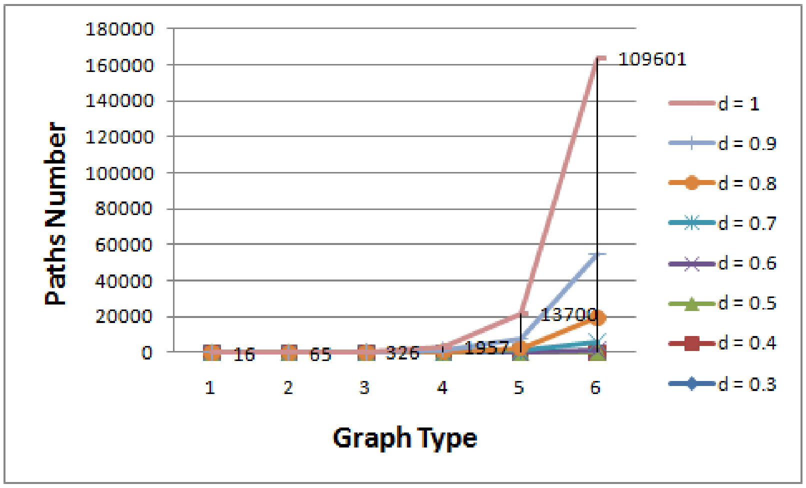


Figure 3.6. Exponential growth nature of MPS enumeration.

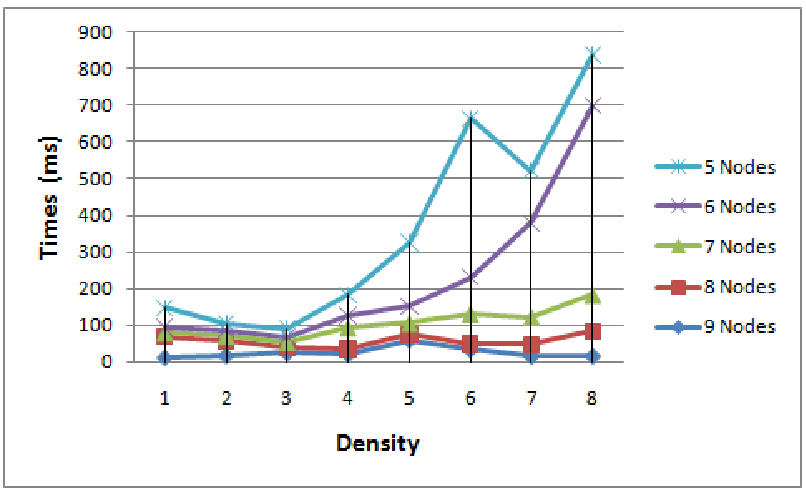


Figure 3.7. Exponential growth nature of MPS enumeration.

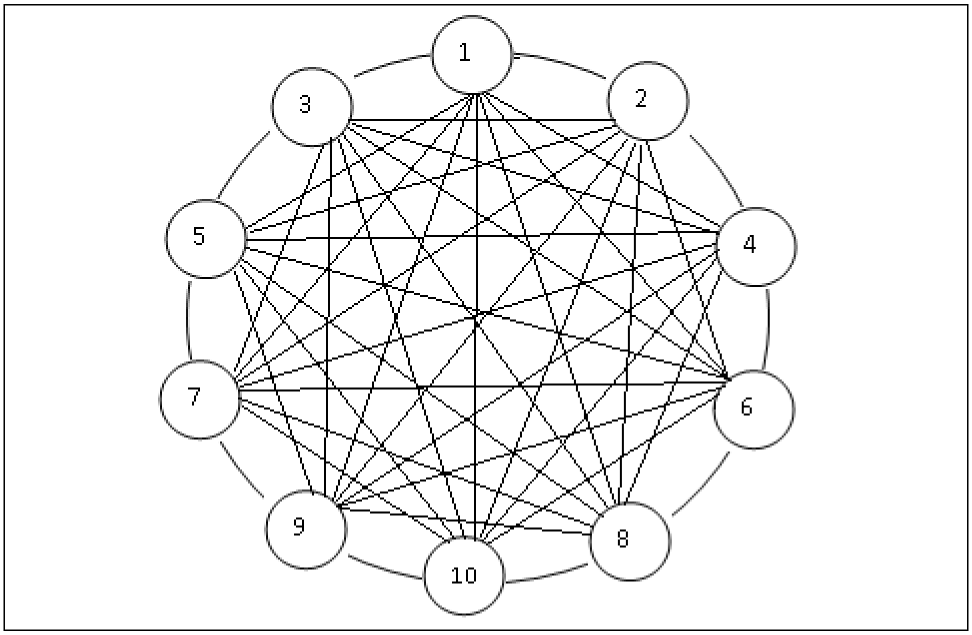


Figure 3.8. A complete network with # of nodes equal to 10.

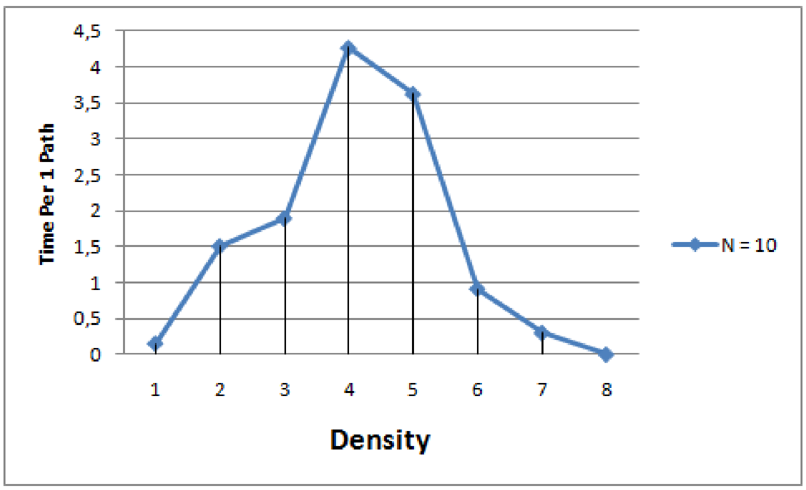


Figure 3.9. Time per 1-path for the graph of figure 3.8.

3.8 Conclusion

In this paper, we have proposed a depth first search-based algorithm to enumerate all minimal pathsets of a network. The program can be used as an input to some tools provided for network reliability measures. The research considers the source-sink problem and can be extended to solve all-reliability and K-reliability cases. The algorithm is finite and the execution time for enumerating MPS of all the networks is very small comparing to some implementations published in the literature. The testing networks are not elementary because the edges are replaced by double arcs which generate a large number of paths. According to the properties of the depth first search we can demonstrate that all the MPS are independents and didn't contain redundant subsections so that to ensure that the MPS are minimal. The comparison with others techniques and others benchmarks gives us an advantage due to the time and space consuming computer capacities. It can be noted from figure 3.9 that the algorithm is robust because computing time by 1-path for dense networks approaches the value zero despite the generated minimal paths number which increases rapidly.

.

REFERENCES

Abraham, J. A., 1979, An Improved Algorithm for Network Reliability, *IEEE Transactions on Reliability,* Vol. R-28, April 1979, pp. 58–61.

Aggarwal, K. K., Misra, K. B. and J. S. Gupta, 1975, A fast algorithm for reliability evaluation, *IEEE Transactions on Reliability*, Vol. R-24, pp. 83-85.

Al-Ghanim, A. M., 1999, A heuristic technique for generating path and cutsets of a general network, *Computers & Industrial Engineering*, Vol. 36, pp. 45–55.

Ball M.O, 1980, Complexity of network reliability computations, Networks, Vol. 19, pp 153-165.

Ball M.O, 1986, Computational complexity of network reliability analysis: An overview, *IEEE Transactions on Reliability*, Vol R35, NO. 3, August, pp 230-239.

Barlow, R. E. and F. Proschan, 1965, Mathematical Theory of Reliability, *J. Wiley & Sons*, 1965, (Reprinted 1996).

Berge, C., 1973, Graphs and Hypergraphs, *North Holland Publishing Company*.

Colbourn, C. J., 1987, The combinatorics of network reliability. New York: *Oxford University Press.*

Heidtmann, K. D., 1989, Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability*, Vol. 38, NO. 3, pp. 305–311.

Jasmon, G. B. & O. S. Kai., 1985, A New Technique in Minimal Path and Cutset Evaluation, *IEEE Transactions on Reliability*, Vol. R-34, NO. 2, pp. 136-143.

Lee. C. Y., 1959, Representation of Switching Circuits by Binary-Decision Programs, *Bell Systems Technical Journal*, Vol. 38, pp. 985–999.

Lin, H. Y., Kuo, S. Y. and Yeh, F. M, 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, *Proceedings of the Eighth IEEE International Symposium on Computers and Communication*, Vol. 2, pp. 1341-1346.

Liu, H. H., Yang, W. T. and Liu, C. C, 1993, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronic Reliability*, Vol. 33, NO. 4, pp. 599-613.

Patvardhan, C., and Prasad., 1996, V. C., Vertex Cutsets of Undirected Graphs, *IEEE Transactions on Reliability,* Vol. 44, N0 2, pp. 347–353.

Rebaiaia, M. L. and Ait-Kadi D. and Merlano, A., 2009, A Practical Algorithm for Network Reliability Evaluation Based on the Factoring Theorem-A Case Study of a Generic Radio communication System, *Journal of Quality*, Vol. 16, N0 5, pp. 323-336.

Shier, D. R. And Whited, A. E., 1985, Algorithms for generating Minimal Cutsets by Inversion, *IEEE Transactions on Reliability,* Vol. R-34, N0 4, pp. 314–319.

Soh, S. and S. Rai, 1993, Experimental results on preprocessing of path/cut terms in sum of disjoint products technique, *IEEE Transactions on Reliability*, vol. 42, no. 1, pp. 24-33.

Tarjan R.E, Depth-First search and linear graph algorithms, 1992, SIAM Journal of Computing, June, pp 146-160.

Yeh, W-C, 2009, A simple Universal Generating Function Method to Search for all Minimal Paths in Networks, *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 39, N0 6, pp. 1247–1254.

Yeh, W-C,2007, A simple Heuristic Algorithm for Generating All Minimal Paths, *IEEE Transactions on Reliability,* Vol. 56, N0 3, pp. 488–494.

Chapter 4

Algorithms for Generating Minimal Cutsets from Binary Decision Diagrams

4.1 ABSTRACT

Network reliability is an important factor which must be evaluated in the design and operation of systems life cycle. This paper presents an efficient method for enumerating minimal cuts of networks. The algorithm proceeds by determining minimal paths set using a fast DFS algorithm and generating minimal cuts set by manipulating binary decision diagrams data structure. The manipulation process consists of a series of transformations, reductions and filtering actions. The correctness of the proposed algorithm is intuitive and easy to be proven. Illustrated examples are proposed to show how to enumerate the network minimal cuts set using the proposed algorithm and an implementation is experienced using a benchmark of many networks.

NOTATION

$G = \ V, E$	Graph network
V	Set of nodes in G
E	Set of edges in G
$n = \|V\|$	Number of nodes in G ($n = \text{cardinal}(V)$)
$m = \|E\|$	Number of edges in G ($m = \text{cardinal}(E)$)
$\bigoplus$	Logical-exclusive or
X	Individual edges (Boolean variable) in G
s, t	Initial and terminal nodes in G
f, g	Boolean formulas
f	Complement of f
x	Complement of the variable x (Boolean)

ACRONYM

$f_{x=1}$	$f_{x=1}$=$f(x_1, x_2, \cdots, 1, \cdots, x_n)\|_{x_i=1}$
$f_{x=0}$	$f_{x=0}$=$f(x_1, x_2, \cdots, 0, \cdots, x_n)\|_{x_i=0}$
PS	Paths set
CS	Cuts set
BDD	Binary decision diagram
OBDD	Ordered BDD
ROBDD	Reduced OBDD
RBD	Reliability block diagram
DFS	Depth first search algorithm.
$f = f(x_1, x_2, \cdots, x_n)$	Boolean expression (function)

NOMENCLATURE

pr_i	Functioning probability of an individual item i (e.g. : a component)
$R(i) = r_i$	Reliability of an item i is the probability of such item to perform its intended function in a specified interval of time $[0, t]$.
$R(G)$	Reliability of a system or a network represented by a graph G.
2-terminal reliability	Reliability that 2 nodes can communicate.
RBD(S)	Graphical representation of the functioning of a system S and its components
MPS	A path P is minimal (MP) if it has no proper sub-paths. MPS is the set of all MP in a graph.
MCS	A cut C is a set of edges such that their removal leads the network to fail. A cuts C set is minimal (MC)if no proper cut exists. A MCS is the set of all MC in a graph.

DEFINITIONS

Shannon decomposition	$f = x.f_{x=1} + \bar{x}.f_{x=0}$
BDD	A BDD is a directed acyclic graph based on Shannon decomposition

4.2 INTRODUCTION

Techniques relative to networks reliability evaluation have been discussed in many publications research and concern a large number of physical systems such as electric power systems, telecommunication networks, traffic and transportation systems, just to name a few (Abraham (1979), Dotson and Gobien (1979), Jasmon and Kai (1985), Heidtmann (1991), Locks and Wilson (1992)). Generally, reliability engineers model the functioning and the physical connectivity of system components by a network. Mathematically, a network is a graph $G(V, E)$ in which the edges E represent the components (e.g. devices, computers, routers, etc.) and the nodes V represent the interconnections. Another representation called RBD is used in theory and has been implemented in some commercial reliability tools in which the components are the nodes and the links are the edges (Figure 4.1 (a)).

Network reliability analysis problem has been the center of many scientific productions. It consists of evaluating the 2-terminal reliability of networks (K and all-terminal) (Hardy et al. (2007), Wood (1986)). General theory, has discussed extensively two techniques; exact (Kuo et al. (2007)) and approximate methods (Lomonosov (1994)). The exact methods employ the concept of MPS/MCS (Yan *et al.*, (1994), Hariri and Raghavendra (1987)). Determining MCS is essential not only to evaluate the reliability indices but also to investigate the different scenarios to find for instance the redundant components which could be replaced to improve the load point reliability. Enumerating all MCS may be a preferable way if the number of paths is too huge to be practically enumerated than the number of cuts. Examples of this kind of preferences is the 2x100 lattice which has 2^{99} paths and just 10000 cuts (Lin et al (2003)), and complete network with 10 nodes from which it can be generated 109601 minimal paths and 256 cuts. In existing algorithms (Yeh (2007)), minimal paths are deduced from the graph using simple and systematic recursive algorithms that guarantee the generated paths set to be minimal. The enumeration of MCS is more problematic because they need advanced mathematics, set theory and matrices manipulation. Many algorithms have been published in the literature. Some of them are implemented in commercial tools. Enumeration appears to be the most computationally efficient. An initiative of solution has been proposed in (Locks (1978)). In this paper Locks presented a method for generating MPS directly from MCS, or vice-versa, for s-coherent systems (Barlow and Proschan (1969)). It starts with the inversion of the reliability expression accomplished by a recursive method combining a 2-step application of De-Morgan's theorems. Yan *et al.* (1994) presented a recursive labelling algorithm for determining all MCS in a directed network, using an approach adapted from dynamic programming algorithms. The algorithm produces all MCS, and uses comparison logic to eliminate any redundant cutsets. This algorithm is an enumeration technique derived from the approach of Jensen & Bellmore (1969) and follows an extension of Tsukiyama *et. al* to improve the computational efficiency and space requirements of the algorithm. Jasmon and Kai (1985) use an algorithm which proceeds by deducting first, the link cutsets from node cutsets and, second the basic minimal paths using network decomposition. So, in addition to the enumeration of cutsets directly, it is possible to obtain them from the inversion of minimal paths (Locks (1978), Shier and Whited (1985)). In such topic, one of the best algorithm is due to Al-Ghanim (Al-Ghanim (1999)) which is based on a heuristic programming algorithm to generate all MPS and Cutsets. The algorithm proceeds by creating a path, then iterates back from an explored node in the current path using unexplored nodes until the source node is reached. Recently, Yeh (Yeh (2007)) presents a simple algorithm for finding all MPS between the source and the sink nodes. It is based on the universal generating function and from which it can be possible to generate MCS. More recently, Rebaiaia and Ait-Kadi (2010), propose an elegant and fast algorithm to enumerate MPS using a modified DFS technique (Tarjan (1992)). The procedure uses each discovered path to generate new MPS from sub-paths. The above procedure is repeated until all MP are found. The algorithm didn't at all produce any redundant MPS. More, they extended their work with theoretical proofs and the usage of sophisticated techniques for dynamic data structures manipulation of complex networks.

This paper presents an efficient technique for determining all MCS of a given directed or undirected graph network using an approach based on BDD's representation. The algorithm proceeds in two stages. First action consists of determining MPS using a fast DFS algorithm. The second action consists of obtaining MCS by manipulating the ROBDD of the MPS. The manipulation process consists of a series of transformations, reductions and filtering actions.

The paper is structured as follows. Section 2 presents some related preliminaries. Section 3 details the principle of BDD manipulations. Section 4 and section 5 give the algorithm and its computational efficiency illustrated using some benchmark networks. The paper concludes the presentation in section 6.

4.3 Preliminaries

Consider a system consisting of m components numbered from 1 to m. Each of these components may be in functioning or failed. Let x_i be the state component and x the state vector, they can be defined as follows :

- $x_i(t) = \begin{matrix} 1 \ \ if\ the\ component\ \ node - link\ \ is\ functioning\ at\ time\ t \\ 0 \ \ if\ the\ component\ \ node - link\ \ is\ failed \end{matrix}$

- $x = (x_1, x_2, \dots, x_m)$ *state vector of the system S of order m such that* $x \in \Omega_i = \{0,1\}^m$ *the state space of the system;*

The system is then represented by its structure function $\Phi: \Omega \to \{0,1\}$.

$$\Phi(x) = \begin{cases} 1 \text{ if the system is functioning when the state vector is } x \\ 0 \text{ if the system has failed when the state vector is } x \end{cases}$$

Systems for which $\Phi(x)$ is a non-decreasing function are called coherent systems [25].

Definition 1. If a system contains P MPS $P_1, P_2, \dots, P_p$ *and C MCS* $C_1, C_2, \dots, C_c$ *its structure function can be represented by :*

$$\Phi(x) = \max_{1 \le j \le P} \min_{i \in P_j} x_i = \min_{1 \le j \le C} \max_{i \in C_j} x_i$$

The reliability R of a system is computed using the following relation :

Consider $E\{\Phi(X)\}$ the mathematical expectation, then :

$$R = Pr\{\Phi(X) = 1\} = E\{\Phi(X)\} = \sum_{X \in \Omega_i} \Phi(X) Pr\{X = x\}$$

and

- $p_i = Pr\{X = 1\}$
- $q_i = Pr\{X = 0\} = 1 - p_i$

After enumerating the MPS/MCS, the reliability evaluation needs the development of the symbolic expression in terms of the probability of the various components being operational/non operational. If the MPS/MCS are mutually exclusive, the probability of the union of *m* events (corresponding to components state; working/failed) can be written if

MPS = $\{P_1, P_2, \dots, P_m\}$ and MCS = $\{C_1, C_2, \dots, C_m\}$ where C_j represents the event "the components of the j^{ith} minimal cut are not functioning", thus

$$\begin{aligned} R\ S\ &= \Pr\ P_1 \cup P_2 \cup \dots \cup P_m\ = \Pr\ P_1) + \Pr\ P_2\ + \cdots + P(P_m \\ &= 1 - (\Pr(C_1) + \Pr(C_2) + \cdots + \Pr(C_n)) \end{aligned}$$

Note that, the last expression is easier to evaluate since it involves a sum of products. Therefore the reliability evaluation is easier if we manage to express the MPS/MCS in a disjunctive form. Knowledge of MPS or MCS allows determining the structure function of any coherent system equivalent to that of the original system, such that the configuration is strictly series or parallel.

For the case of complex system $P_1, P_2, \dots, P_i$ are not necessary to be expressed in disjunctive form. They can be transformed using the following relation :

$$\Pr\ P_1 \cup P_2 \cup \dots \cup P_m\ = \Pr\ P_1\ + \Pr(P_1 P_2) + \Pr(P_1 P_2 P_3) + \cdots + \Pr(P_1 \dots P_{i-1} P_i)$$

4.3.1 Illustrative example

Consider a directed *bridge* network represented in Figure 4.1 (b). The MPS and MCS are :

MPS : {P_1 = {x_1, x_4}; P_2 = {x_2, x_5} and P_3 = {x_1, x_3, x_5}}.
MCS: {C_1 = {x_1, x_2}; et C_2 = {x_1, x_5}; C_3 = {x_4, x_5}; and C_4 = {x_2, x_3, x_4}}.

The structure function of the system is equivalent to the series-parallel graph (Figure 4.1 (c)). It can be written as follows :

- Case MPS (Figure 4.1.c):

$$\Phi\big(X(t)\big) = 1 - (1 - x_1x_4)(1 - x_2x_5)(1 - x_1x_3x_5)$$

and the reliability is :

$$R\ \ t\ \ = \Pr\ \ \Phi\ \ X\ \ t\ \ \ = 1$$

$$= r_1 r_4 + r_2 r_5 + r_1 r_3 r_5 - r_1 r_2 r_4 r_5 - r_1 r_3 r_4 r_5 - r_1 r_2 r_3 r_5 + r_1 r_2 r_3 r_4 r_5$$

- Case MCS (Figure 4.1. D):

$$\Phi\ \ X\ \ t\ \ \ = \ \ 1 - x_1\, x_4\ \ \ 1 - x_2\, x_5\ \ \ 1 - x_1\, x_3\, x_5$$

$$= \ \ 1 - x_1\, x_4\ \ \ 1 - x_1\, x_3\, x_5 -\ x_2\, x_5 + x_1\, x_2 x_3\, x_5$$

$$= \ \ 1 - [x_1\, x_3\, x_5 +\ x_2\, x_5 - x_1\, x_2 x_3\, x_5 + x_1\, x_4 - x_1\, x_3 x_4\, x_5 - x_1\, x_2 x_4\, x_5 + x_1\, x_2\ \ x_3\, x_4\, x_5]$$

$$R\ \ t\ \ = \Pr\ \ \Phi\ \ X\ \ t\ \ \ = 1$$

$$= \ \ 1 - [r_1\, r_3\, r_5 +\ r_2\, r_5 - r_1\, r_2 r_3\, r_5 + r_1\, r_4 - r_1\, r_3 r_4\, r_5 - r_1\, r_2 r_4\, r_5 + r_1\, r_2\ \ r_3\, r_4\, r_5]$$

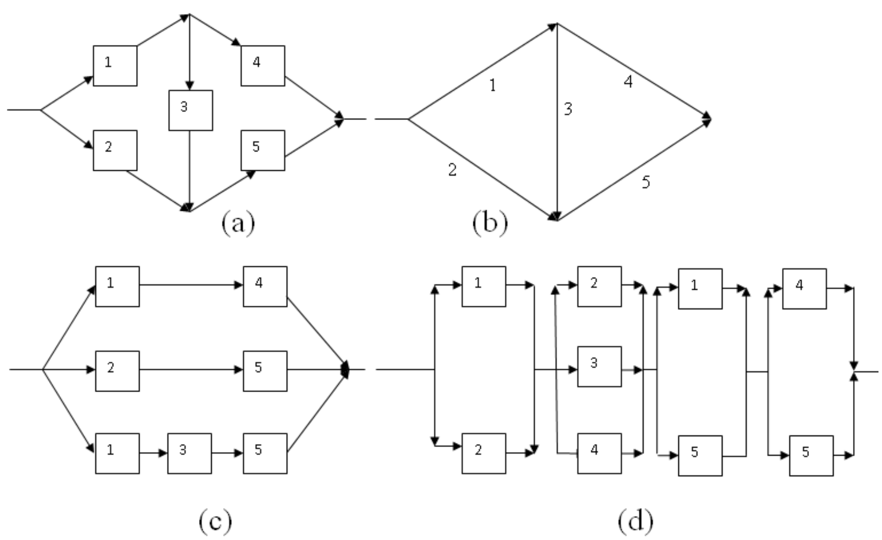


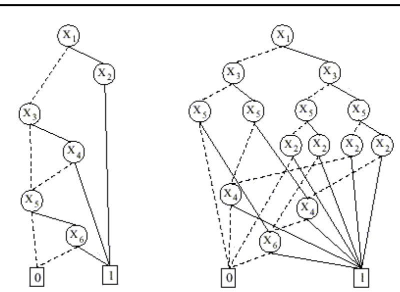
Figure 4.1. a) System structure b) Reliability network c) Reliability structure based on (a): MPS. d) Reliability structure based on (a): MCS.

4.4 BINARY DECISION DIAGRAMS SIMPLIFICATION APPROACH

Boolean algebra reduction is not sufficient when the structure function is complex. Many techniques have been found since the sixties but also they still not adequate for simplifying the combination of MPS/MSC equations. Solution of such problem has been solved partially by discovering new representations of Boolean relations based on the theorem of Shannon. Bryant (1986) was the first to use the work of Akers (Akers, 1978) on the use of binary decision diagrams for symbolic verification of integrated circuits. The implementation and manipulation of BDD algorithms is composed by three procedures, *restrict*, *apply* and *ite*. They have been first investigated and implemented by Bryant (Bryant, 1986), (Bryant, 1992). The problem with BDD representation despite their effectiveness is that, their exponential growing size due to a wrong order declaration between variables. Ruddell (Rudell, 1993) first used an algorithm based on dynamic programming techniques to reduce the size of the BDD and Bollig *et al.* (1996), demonstrate that improving the Variable Ordering of OBDD is NP-Complete. BDD principle has been used in many fields for simplifying Boolean expression like electronic circuits design, formal methods for model-checking. Coudert and Madre (1992) and Rauzy (1993) applied first, BDDs for evaluating networks reliability.

The representation and the simplification of a Boolean expression proceeds in 4-steps:

- Construct the binary decision tree (BDT) associated with the graph formula.
- Transform the BDT to a BDD by applying the following rules (see Figure 4.2) by :
    - a- Merging equivalent leaves of a binary decision tree.
    - b- Merging isomorphic nodes.
    - c- Elimination of redundant tests
- Transform the BDD to OBDD by just a wise choice on variables (see Fgure 4.2 for an example of a good decision).



$x_1 < x_2 < x_3 < x_4 < x_5 < x_6$; $x_1 < x_3 < x_5 < x_2 < x_4 < x_6$

Figure 4.2. Two ordering of the same expression

- OBDD can be reduced to a ROBDD by repeatedly eliminating in a bottom-up fashion, any instances of duplicate and redundant nodes. If two nodes are duplicates, one of them is removed and all of its incoming pointers are redirected to its duplicate. If a node is redundant, it is removed and all incoming pointers are redirected to its just one child (see figure 4.3).

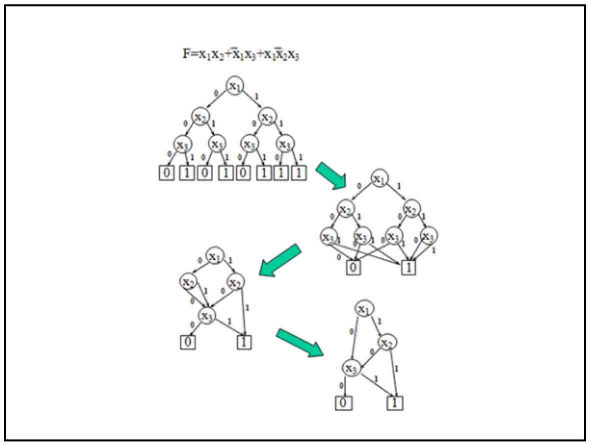


Figure 4.3. BDD reduction steps

To overcome this deficiency, Bryant (1986) suggested a new way to represent the decomposition procedure of Shannon (Lee , 1959), called *ite* for (If _ Then _ Else), which in turn is expressed by

$f = ite\ \ x, F_1, F_2\ \ = \ x.F_1 + x.F_2$; with $F_1 = f_{x=1}$ and $F_2 = f_{x=0}$.

ite procedure receives OBDDs for two Boolean formulas F_1 and F_2 for $F_1\ \ OP\ \ F_2$ such as every Boolean expression could be written using the *ite* function and OP is a boolean operator. Figure 4.4, show some elementary formulas with their corresponding *ite* expressions. The *ite* algorithm maintains a computed table (Figure 4.4. (left)), which memorize the intermediary calculus to avoid determining the same expression repeatedly and a unique table (Figure 4.4 (right)). The unique table is a dynamic matrix which assigns a line to represent a node and the columns represent the node identifier, the name of the associated variable, left son identifier and right son identifier.

Boolean formulas	Ite form
0	0
1	1
$f.g$	$ite(f,g,0)$
$f.g$	$ite(f,g,0)$
$f.g$	$ite(f,0,g)$
$f\oplus g$	$ite(f,g,g)$
$f+g$	$ite(f,1,g)$
$f+g$	$ite(f,0,g)$
f	$ite(f,0,1)$
$f+g$	$ite(f,1,g)$
$f+g$	$ite(f,g,1)$
$f.g$	$ite(f,g,1)$

	0	-	
	1		
2	a	3	4
3	b	5	6
4	b	7	8
5	c	0	0
6	c	0	1
7	c	1	0
8	c	0	1

Figure 4.4. Left: Table of the simplified Boolean functions presented as an *ite* relation. Right : Representation of BDD in computer.

The following pseudo-code gives the *ite* function.

```
Function ite(f, g, h)
   if f = 0 then
       Return h;
   else if f = 1 then
           Return g;
        else if (g = 1) ∧ (h = 0) then
           Return f;
            else if g = h then
              Return g;
                else if ∃ computed-table entry (f, g, h,H) then
                  Return H;
                     end if
                 x_k ←top variable of f, g, h;
                 H ← new non-terminal node with label x_k;
               then H ← ite (f |x_k=1, g |x_k=1, h |x_k=1);
                  else H ← ite (f |x_k=0, g |x_k=0, h |x_k=0);
        Reduce  H;
 Add entry (f, g, h,H) to computed-table;
        Return  H;
 end.
```

Figure 4.5. ITE algorithm

APPLY procedure is an efficient tool to combine functions using binary operators like the conjunction and disjunction Boolean operators. It is the major core of our algorithm. It can also be used to complement a function; it suffices in such case to complementing the values of terminal vertices.

According to Bryant, the APPLY procedure takes graphs representing functions f_1 and f_2, a binary operator (say OP) and produces a reduced graph representing the function $f_1 \ \ OP \ \ f_2$. It proceeds as follows:

Consider graph trees of two functions f_1 and f_2 (e.g. $f_1 = ac$ and $f_2 = bc$ (see figure 6)), and suppose that $v1$ and $v2$ are respectively their roots. The Apply algorithm composes two Boolean formulas as follows :

Procedure *Apply*(f_1, f_2, OP, f)

If both $v1$ and $v2$ are terminal vertices, *then* the resulted graph consists of a terminal vertex having *value* ($v1$) <*op*> *value* ($v2$); *else* *If* $v1$ or $v2$ non terminal vertex, *then*

if index ($v1$) = *index* ($v2$) = *i*
create a vertex *u* having index *i*
apply the algorithm recursively on *low* ($v1$) and *low* ($v2$) *then*
generate the subgraph whose root becomes *low* (*u*),
apply the algorithm recursively on *high* ($v1$) and *high* ($v2$)
generate the subgraph whose root becomes *high* (*u*)

else

If index ($v1$) = *i*, then

if $v2$ is a terminal vertex or *index* ($v2$) > *i then*
create a vertex *u* having index *i*,
apply the algorithm recursively on *low* ($v1$) and $v2$
generate the subgraph whose root becomes *low* (*u*)
apply the algorithm recursively on *high* ($v1$) and $v2$
generate the subgraph whose root becomes *high* (*u*).

end_procedure

Figure 4.6. Apply procedure for composing two formulas.

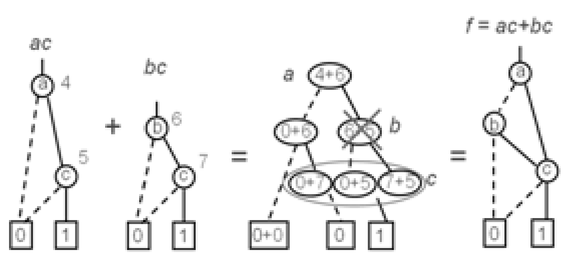


Figure 4.7. Composition Example.

4.5 GENERATION OF ROBDD FROM MPS

Consider the bridge graph (figure 4.1 (a)) and its generated minimal paths such as:

MPS : $\{P_1 = \{x_1, x_4\}; P_2 = \{x_2, x_5\}; P_3 = \{x_1, x_3, x_5\}\}$.
We use the procedure Apply for composing the above three minimal paths. Thus, the algorithm proceeds as follows:

Algorithm *Compose* (MPS, matrice_container)

Construct the BDD of MP P_1 (Figure 4.9 (a)).

1- Construct the BDD of MP P_2 (Figure 4.9 (b)).

2- Compose the BDD of MP P_1 and the BDD of MP P_2 (Apply procedure). Let P such BDD.

3- Construct the BDD of MP P_3 (Figure 4.9 (d)).

4- Compose the BDD of MP P and the BDD of MP P_3. Let PP such BDD (Apply procedure).

5- Recover the structure information's of the BDD resultant and put it in matrix container (Figure 4.9 (b)).

6- end_procedure

Figure 4.8. Algorithm Compose

The following graphs give the composition based-BDD of bridge network MPS.

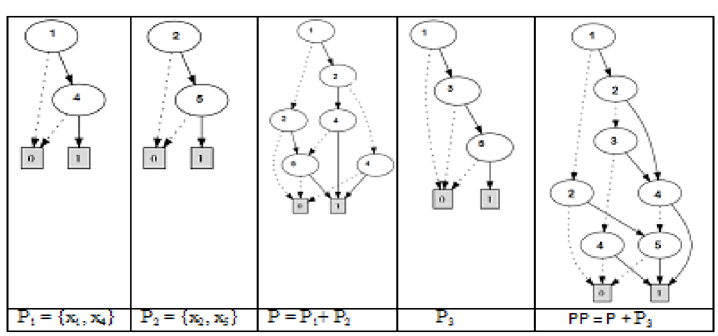
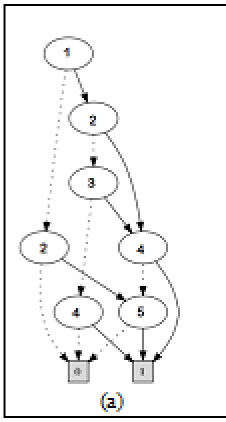


Figure 4.9. Application of the composition procedure (Apply) using MPS of bridge Network



[8]	4:	0	1
[10]	5:	0	1
[8010]	4:	10	1
[8011]	3:	8	8010
[8013]	2:	0	10
[8014]	2:	8011	8010
[8015]	1:	8013	8014

(b)

Figure 4,10. (a) : ROBDD network of figure 1.(a) and (b) : its representation code in memory similar to table in figure 5 (right).

4.6 Generation of MCS from MPS ROBDD

As described by Locks (Locks, 1978) and, Shier and Whited (1985), an inverse polynomial to the path polynomial can be obtained by complementing the given polynomial and using DeMorgen's laws. The complementation of the above MPS for the bridge network (Figure 4.1 (a)) results in :

$$x_1 + x_4 \quad x_1 + x_3 + x_5 \quad x_2 + x_5 \quad = x_1 \ x_2 + x_1 \ x_3 \ x_5 + x_4 \ x_5 + x_2 \ x_3 \ x_4$$

If we transpose the idea of generating MCS by inversion as introduced by Locks (1978) but not directly on Boolean formulas polynomial, we can use the model generated by the graphical representation of BDD (e.g. Figure 4.8. (a)) instead of working directly on formulas manipulation.

The new idea is explained first using the bridge network to show how one can generate minimal cuts by just using a DFS procedure (Tarjan, 1972)), which visits BDD graph nodes representing the Boolean decision variables and edges.

Consider the bridge network given in Figure 4.1. (a), its corresponding ROBDD constructed from the MPS is shown in Figure 4.8. (a). The cuts set are found from branches on this tree by tracing in a reverse sense from button to top that is to say from square node 0 to the variable on the root. The second phase proceeds by removing from the branches cuts. In the third phase, the algorithm deletes all redundant cuts to build the MCS. The procedure to deduce the MCS is a depth first search algorithm. It works on the graph using data information's taken from matrix of the Figure 4.8. (b). Il can be presented as follows:

Procedure *Generation_of MCS*

- Place the squared node on top of a stack 1 //* records DFS visits to ROBDD nodes *//.
- Place the squared node on top of a stack 2 //* records cut's nodes.
- Place on the top of the stack 1 all the ascending nodes of the top variable in the stack.
- Place the node top of the stack 1 on top of the stack 2, if the edge (link) is dotted.
- Continue until the variable reach the root node.
- If so, a cut has been found. Write the content of the stack 2 as a line of a matrix. Remove top variable from stack 1 and from stack 2.
- Continue the procedure until stack 1 is empty.
- Apply the filtering process by removing all the redundant paths (cuts) using the matrix of paths (cuts)
- Display MSC Matrix.

end_procedure

Figure 4.11. Procedure to generate MCS

The filtering procedure removes redundant cuts from the set of all the cuts. It proceeds as follows:

1- Sort the matrix CS in an ascending order according to the size of each vector (number of variables);

2- Take the first vector and compare it with each of the others vectors;

3- If the members of the intersection are equal to the first vector then remove the actual vector from CS matrix;

4- Iterate using the others vectors of the matrix.

The following pseudo-description shows the processing of the filtering procedure as explained bellow:

```
Procedure filtering(CS, MCS)

   n = length(CS); /* size of matrix vector */
   m = size(CS);  /* size of matrix vector
  for i = 1,m-1
     v(k) = CS(i,k) (k=1,...,n)(CS(i,k)≠0)
    for j = i+1,m
       w(k) = v(k) ∩ CS(j,k) (k=1,...,n)(CS(i,k)≠0)
       if w(k) = v(k) (CS(j,k) is a redundant vector)
          remove vector  CS(j,k);
       end_if
    end_for
 end_for
   MPS = CS
   display MPS
end_procedure
```

Figure 4.12. Filtering algorithm for MCS generation

6.6.1 Illustrative example. (ROBDD of the bridge network MPS)

Consider the graph illustrated in figure 4.10.(a).

Let's start the search from the leaf squared with the value 0. We'll go from node to node via dotted edge (value of the immediate node variable = 0). If we take any branch from the leaf-0 to the root node, we construct a path with the concatenation of the variables on the nodes. Thus, all the branches will generate all cuts set CS. We remark that cuts and paths are complementary. Table 4.1 shows the results when applying the procedure Generation_of MCS. Note that the intermediary results on the stack have been omitted from their presentation in the following tables.

Table 4.1. Deduction of all paths from graph in Figure 4.7.

Cuts	Branches	All CS
1st	leaf 0- dotted edge - x_2- dotted edge – x_1 (stop).	x_1, x_2
2nd	leaf 0- dotted edge -x_4- dotted edge-x_3-dotted edge-x_2- solid edge x_1 (stop).	x_2, x_3, x_4
3nd	leaf 0- dotted edge -x_5- solid edge-x_2- dotted edge-x_1 (stop).	x_1, x_5
4nd	leaf 0- dotted edge -x_5- dotted edge-x_4- solid edge-x_3-dotted edge-x_2- solid edge x_1 (stop)	x_2, x_4, x_5
5nd	leaf 0- dotted edge -x_5- dotted edge-x_4- solid edge-x2-solid edge- x_1 (stop).	x_4, x_5

The filtering procedure is explained using the results of table 4.1. The procedure compares the MCS vectors and removes those who contain duplicated variable as shown in table 4.2 (dashed variable are redundant variable). Vector which contains redundant variables is removed from the MCS list (set).

Table 4.2. All CS and MCS after filtering

All CS	All CS sorted	Not minimal	MCS
x_1, x_2	x_1, x_2		x_1, x_2
x_2, x_3, x_4	x_1, x_5		x_1, x_5
x_1, x_5	x_4, x_5		x_4, x_5
x_2, x_4, x_5	x_2, x_3, x_4		x_2, x_3, x_4
x_4, x_5	x_2, x_4, x_5	x_2, x_4, x_5	

6.7 EXPERIMENTAL RESULTS

The proposed algorithms and procedures have been implemented in MatLab 8 and Java Jdk 1.6. A communicating interface has been written to render easy data and results transfer between MatLab system and Java packages running under jGrasp a graphical tool written in Java JDK. The operating system is 32 bits and 2038 MO of Windows Vista of Microsoft. The machine is an HP notebook PC with an Intel(R) core (TM) 2 Duo processor of 1.67. The benchmark networks in figure 9 were used and the results are shown in table 4. All the networks are 2-terminal and they have been used in different publication papers. We can remark from table 4, that the value of execution time is interesting despite the fact that the performance of the machine characteristics is not high. The importance of this work shows the efficiency of the algorithms. Note that no comparison was made with another implementation but they can be compared for example with the results of Lin *et al.* (2003). It is certain that if the CPU was more powerful and memory space was wider one can easily handle more complex networks.

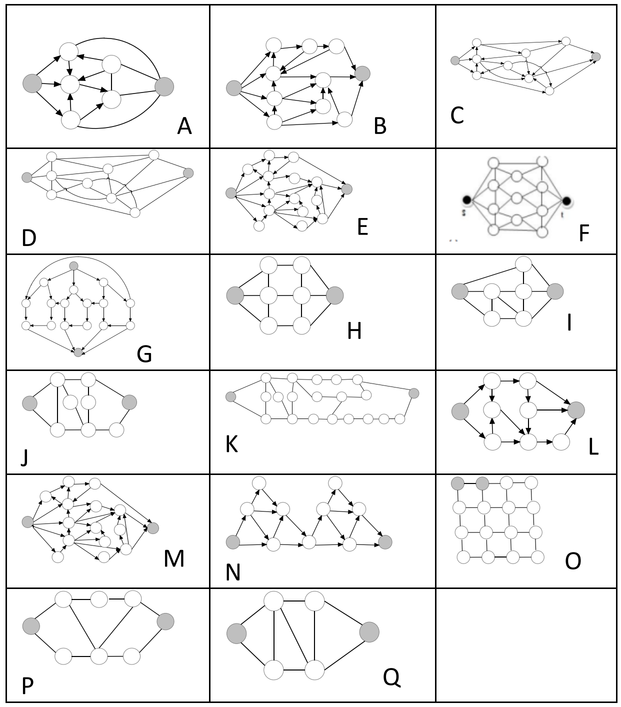


Figure 4.13. Benchmark networks

Table 4.3. Benchmark results for 2-terminal networks

Networks	MPS	MCS	Time(s)
A	8	12	0.075591
B	18	110	0.282727
C	115	85	313.17
D	33	72	11.29
E	35	3	0.046
F	114	562	21236.06
G	10	959	818.80
H	29	29	0.708676
I	25	20	0.332611
J	13	21	4.059842
K	44	528	2572.03
L	6	23	0.226500
M	36	96	11.166
N	100	16	8.3297
O	98	105	283.38
P	5	16	0.768533

Q	13	9	0.945836

6.8 CONCLUSION

Enumerating minimal cut sets constitutes an important element for evaluating the load point reliability indices but also it is used to investigate solutions for optimizing networks performance based on indications showing the components which can be improved by redundancy for augmenting the network reliability.

This paper introduces a simple approach based on binary decision diagrams representation and Booleans simplification. The paper shows by the application of a series of algorithms and procedures all the phases of transformation to get at the end a Matrix representing minimal cut set. The method is similar to those of Locks technique but its efficiency is that it works on large networks.

REFERENCES

Abraham J. A., 1979, An Improved Algorithm for Network Reliability, *IEEE Transactions on Reliability*, pp. 58–61.

Akers. S. B., 1978, Binary Decision Diagrams, *IEEE Transactions on Computers*, C-27(6), pp. 509–516.

Al-Ghanim A. M., 1999, A heuristic technique for generating minimal path and cutsets of a general network, *Computers & Industrial Engineering*, Vol. 36, pp. 45–55.

Barlow R. E. and F. Proschan, 1965, Mathematical Theory of Reliability, *J. Wiley & Sons*, (Reprinted 1996).

Bollig B., Ingo W., 1996, Improving the Variable Ordering of OBDDs Is NP-Complete, *IEEE Transactions on Computers*, Vol. 45, NO. 9, pp. 993–1002.

Bryant. R. E., 1986, Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers*, C-35(8), pp. 677–691.

Bryant. R. E., 1992, Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams, *ACM Computing Surveys*, Vol. 24, No. 3, pp. 293–318.

Coudert O. and J. C. Madre, 1992, Implicit and incremental computation of primes and essential primes of Boolean functions, in *Proc. of the 29th ACM/IEEE Design Automation Conference*, IEEE Computer Society Press, pp. 36-39.

Dotson W. P., J. 0. Gobien., 1979, A new analysis technique for probabilistic graphs, *IEEE Trans. Circuits and Systems*, Vol-Cas 26, No. 10, pp. 855-865.

Hardy. G., C. Lucet, and N. Limnios., 2007, K-Terminal Network Reliability Measures With Binary Decision Diagrams, *IEEE Transactions on Reliability*, Vol. 56, NO.3, pp. 506-515.

Hariri, S., and Raghavendra, Syrel: A symbolic reliability algorithm based on path and cutest methods, *IEEE Transactions on Reliability*, C-36(10), pp. 1224-1232.

Heidtmann K.D., 1991, Smaller Sums of Disjoint Products by Subproduct Inversion, *IEEE Transactions on Reliability*, Vol. 38, pp. 305-311.

Jasmon, G. B., and Kai O. S.., 1985, A New Technique in Minimal Path and Cutset Evaluation, *IEEE Transactions on Reliability*, Vol. R-34, NO. 2, pp. 136-143.

Jensen P., M. Bellmore, 1969, An algorithm to determine the reliability of a complex system, *IEEE Transactions on Reliability*, Vol R-18, NO. 1, pp. 169-174.

jGRASP, Auburn University. http://www.jgrasp.org.

Kuo S. Y, Yeh., F.M, and Lin.,Y.H., 2007, Efficient and Exact Reliability Evaluation for Networks With Imperfect Vertices, *IEEE Transactions on Reliability*, Vol. 56, No 2, pp. 288-300.

Lee. C. Y., 1959, Representation of Switching Circuits by Binary-Decision Programs, *Bell Systems Technical Journal,* Vol. 38, pp. 985-999.

Lin, H. Y., Kuo, S. Y. and Yeh, F. M, 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, in *Proc. of the Eighth IEEE International Symposium on Computers and Communication*, Vol. 2, pp. 1341-1346.

Locks, M. O. and Wilson, J. M, 1992, Note on disjoint products algorithms, *IEEE Transactions on Reliability*, Vol. 41, No 1, pp. 81-84.

Locks, M., 1978, Inverting and minimalizing path sets and cutsets, *IEEE Transactions on Reliability*, Vol R-27, Jun, pp. 107-109.

Lomonosov M., 1994, On Monte Carlo estimates in network reliability, Probability in the Engineering and Informational Sciences, Vol. 8, pp. 245-264.

Mohamed-Larbi Rebaiaia, Daoud Ait-Kadi, 2010, A Program for Computing the Reliability of Communication Networks, *ACCA IFAC, IEEE Conference LACC Latin Americain Control Conference Chili*, August, pp. 24-27.

Rauzy, A., 1993, New Algorithms for fault tree analysis, Reliability Eng. and Systems Safety, vol. 40, pp. 203-210.

Rudell, R, 1993, Dynamic variable ordering for ordered binary decision diagrams, in *Proc. of the 1993 IEEE/ACM international conference on Computer-aided design*, pp. 42-47.

Shier D. R., D.E. Whited, 1985, Algorithms for generating minimal cutsets by inversion, *IEEE Transactions on Reliability*, vol R-34, pp 314- 319.

Tarjan, R, 1972, Depth-first search and linear graph algorithms, *SIAM Journal of Computing*, Vol. 1, No 2, pp. 146-160.

Theolougou R., J.G. Carlier, 1991, Factoring & Reduction for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 40, No 2, pp. 210-217.

Tsukiyama, S.I., Shirakawa, H. Ozaki, H. Ariyoshi, 1980, An algorithm to enumerate all cutsets of a graph in linear time per cutset", *Journal of the Assoctatlon for Computing Machmery*, Vol 27, No 4, pp 619-632.

Wood R. K., 1986, Factoring algorithms for computing K-terminal network reliability, *IEEE Transactions on Reliability*, R-35, 269-278.

Yan, L., Taha, H., and Landers, L. L., 1994, A recursive Approach for Enumerating Minimal Cutsets in a Network, *IEEE Transactions on Reliability*, Vol 43, No 3, 383-388.

Yeh W-C, 2007, A Simple Heuristic Algorithm for Generating All Minimal Paths, *IEEE Transactions on Reliability*, Vol. 56, NO 3, pp. 488-494.

Chapter 5

A Contribution for Computing Networks Reliability using Reduced Binary Decision Diagrams

5.1 ABSTRACT

The reliability of networks is defined as the probability that a network will perform its intended function without failure over a given period of time. Computing the reliability of networks is an NP-hard problem, which need efficient techniques to be evaluated. This paper presents a network reliability evaluation algorithm using Binary Decision Diagrams (BDD). The solution considers the 2-terminal reliability measure and proceeds first by enumerating the minimal paths set using a recursive depth first search procedure from which a BDD is obtained. The algorithm has been implemented on a Personnel computer and didn't require large memory size and time requirement for average size graphs.

5.2 INTRODUCTION

A probabilistic graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ is a finite set *V* of nodes and a finite set *E* of incidence relations on the nodes called edges. The edges are considered as transferring a commodity between nodes with a probability $\boldsymbol{p}$ called reliability. They may be directed or undirected and are weighted by their existence probabilities. The graph in such case, models a physical network, which represents a linked set of components giving services. The reliability of networks is defined as the probability that systems (networks) will perform their intended functions without failure over a given period of time. Figure 5.1 shows an example of an undirected graph.

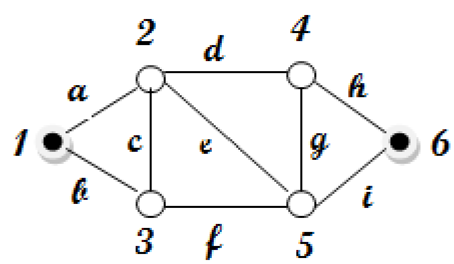
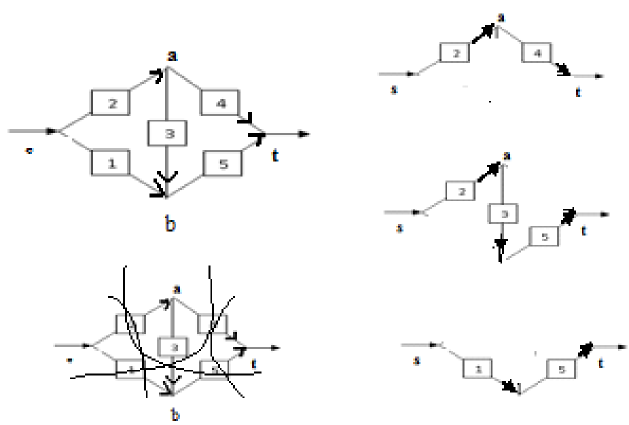


Figure 5.1. A probabilistic weighted graph with six nodes (*1, 2, 3, 4, 5* and *6*) and nine undirected edges (*a, b,...,i*).

This chapter presents an algorithm for determining the reliability of a given network G when one node is identified as the source user s and another as the terminal user *t*. The terminal reliability $R(G)$, is defined as the probability that at least one path will exist from s to *l*.

Techniques for determining the reliability of networks are classified as enumerative techniques, inclusion/ exclusion, and SDP for sum of disjoints products (Locks *et al.* (1992)), and those based on the well-known network decomposition (Dotson *et al.* (1979), Theologou *et al.*(1991), and Wood (1986). Others techniques like simulation for example are considered as approximate methods (Fishman (1986)). In the former techniques, the problem consists of determining minimal paths (respectively Cuts) between *s* and *t* given the probability of success for each communication link (edge) in the network. By definition, a MPS (resp. MCS) is a path (respectively a cut) from which it is impossible to extract another path (respectively a Cut). Figure 2 shows MPS and MCS and the relative matrices of a directed graph.



$$MPS = \begin{matrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{matrix} ;$$

$$MCS = \begin{matrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}$$

Figure 5.2. Network with its 3 MPS (right) and 4 MCS (down at left).

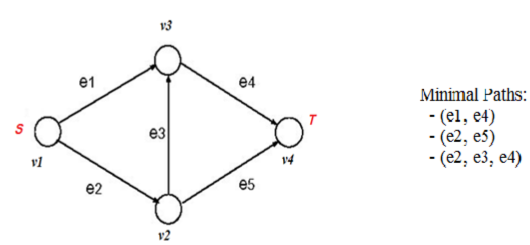
The concept of using minimal paths set (MPS) and minimal cuts set (MCS) to analyze probabilistic graphs appears to have been first explored by several authors (Abraham (1979), Jasmon et al (1985), Locks (1992)). Many algorithms have been proposed to generate MPS/MCS for directed or undirected graphs. Some of the algorithms need special data preparations and require advanced mathematics (Yan *et al.* (1994)). In Shier *et al*. (1985), Locks proposed a simple method to generate MCS directly by inverting MPS. Such solution has been soon discarded because it may be impractical for generating MCS in the presence of large network. Hariri *et al*. (1987) give a simple and efficient algorithm, SYREL, to obtain compact terminal reliability expressions. This algorithm incorporates conditional probability, set theory, and Boolean algebra. As the evaluation of the reliability is an NP-hard problem, so approximation becomes a good solution for large systems. Another approach uses BDDs (binary Decision Diagrams) Bryant (1986) and Bryant (1992). Because BDDs and reliability factoring representation are formalized from the well-known Shannon decomposition theorem, they are more convenient to derive an algorithm to compute the reliability especially for networks with topologies that contains isomorphic subgraphs. Their application in the reliability analysis framework has been introduced by Coudert (1992) and extended to fault trees by Rauzy (1992). Recently, many papers published in reliability literature propose new algorithms based on Reduced ordered BDD (ROBDD). Hardy *et al.* (2007) and Lin *et al.* use ROBDD to derive interesting fast solutions for computing reliability of large systems such the 2×100 lattice network which has 2^{99} paths (Lin *et al.* (2003), and Shier (1985)).

In the following, we propose an efficient algorithm to evaluate the 2-terminal network reliability. The algorithm utilizes minimal paths set generated using one of the best and fast methods developed by us. The method uses a simple technique similar to the depth first search algorithm of Tarjan (1972), which explores the graph by crossing nodes and edges from the top to the bottom and backtracks until all the nodes are marked. Each time a minimal path is found a symbolic reliability function is generated by composing the last BDD with the actual one corresponding to the generated path. A new BDD is memorized for the next iteration until no path is generated. At each composition, some operations are applied to reduce considerably the size of BDD by merging graphs and structure reductions.

The chapter is organized as follows: Section 2 presents some preliminaries concepts such as network and *s-t* terminal reliability evaluation. Section 3, gives the description of minimal paths set and presents an algorithm to enumerate them. In section 4, we present theorems relative to binary decision diagrams representation and theirs extensions to cover network reliability class of problems. In section 5 we detail some experiment benchmarks to check the efficiency of BDD formalism. Finally some concluding remarks are given in section 6.

5.3 NETWORKS MODELING

Consider a network $G = (V, E)$ as discussed below. For a specified set of nodes $K \subseteq V$ of G, we denote the K-terminal reliability of G by $R(G_K)$. When $|K| = 2$, $R(G_K)$ is called 2-terminal (or terminal-pair) reliability which defines the probability of connecting the source node with a target node. A success set, is a minimal set of the edges of G such that the vertices in K are connected; the set is minimal so that deletion of any edges causes the vertices in K to be disconnected and this will invalidate the evaluation of the reliability. Topologically, a success set is a minimal tree of G covering all vertices in K. The computation of the K-terminal reliability of a graph may require efficient algorithms. One such solution can be derived directly from the topology of the network by constructing a new parallel-series network using MPS of the original network such that each minimal path constitutes a branch of the parallel-series graph. Then, a characteristic expression $\Phi \; X \; t$ is derived from the disjoint expressions of paths terms, and from which the reliability is evaluated after applying Boolean simplification processing. Figure 5.3, gives an example and the expression of its reliability. Note that $X \; t$ is Boolean state variables vector and each r_i is the value of the reliability of the component (edge) i which replaces the state variable x_i after the reductions.



$$\Phi(X(t)) = 1 - (1 - x_1x_4)(1 - x_2x_5)(1 - x_2x_3x_4).$$

$$R_{1\text{-}4} = Pr\,\{S_{1\text{-}4}\} = \Pr(\Phi(X(t)) = 1) = r_1 r_4 + r_2 r_5 + r_2 r_3 r_4 - r_1 r_2 r_3 r_4 - r_2 r_3 r_4 r_5 - r_1 r_2 r_4 r_5 + r_1 r_2 r_3 r_4 r_5$$

Figure 5.3. Simple Model for the evaluation of network reliability

5.4 PROCEDURE FOR MINIMAL PATHS SET ENUMERATION

Many algorithms have been proposed to generate MPS for directed and undirected graphs. Some algorithms are more difficult to be implemented because they require the manipulation of mathematical operations on large matrices.

In the following we present a fast procedure to deduce minimal paths set. The kernel of the procedure uses a recursive function similar to the depth first search algorithm of Tarjan (Tarjan (1972)).

Note: The complexity of the depth first search algorithm is $O(|V| + |E|)$.

The following algorithm will be executed to find minimal paths, so that each execution cycle will correspond to a minimal path. This procedure is called recursively as many times as all the nodes have not been visited and the edges not marked. The algorithm is shown in the pseudo-code form in figure 5.4 and it is invoked inside the main program in which Stacks, initial and terminal nodes are initialised with values equal to zero.

```
Algorithm stack S = pathDFS(G, v, z)
setLabel(v, VISITED)
S.push(v)
 if   v = z
 return S.elements()
 for all e in G.incidentEdges(v)
if getLabel(e) = UNVISITED
w ← opposite(v,e)
 if getLabel(w) = UNVISITED
```

```
S.push(e)
pathDFS(G, w, z)
S.pop(e)
 Else
S.pop(v)
end
Program Main()
Input G     // Reading the matrix of the graph G as link by link (link : initial
node; terminal node)
Declaring dynamics vectors and stacks (put in them zeros)
Declaring initial and terminal nodes (v, z)
Do While .true.
pathDFS(G, v, z)
if "the last minimal path have been encountered"
return .false.
enddo
```

Figure 5.4. Minimal paths set enumeration algorithm.

The algorithm sets the three stacks to zero, one is used for memorizing the minimal path, the second keeps the traces of the encountered nodes and the third marks the edge position in the path. Then the algorithm begins by iterating. First it determines the children of the initial node, and at each time the comparison of the actual node with the terminal permits to decide if a minimal path has been generated. If so, the stack memorizing the path is transferred to the BDD module and the algorithm backtracks using the other node son. If it exists, it forwards until to reach the terminal node. The algorithm continues by backtracking and forwarding operations until no node is present in the corresponding stack. At the end, all the paths have been passed to the BDD module and a general expression ROBDD of the reliability is generated from which the reliability of the network is computed.

5.4.1 Step-by-Step Example: Bridge network

In the following, a simplified example shows the application of the algorithm for enumerating all the MPS of the network presented in Figure 5.2.

The computing program begins first by reading the matrix associated to the graph. Value 1, represents the existence of an edge (arc) between two nodes and zero the opposite case.

In the first step, the algorithm gives a series of successive numbers in an increasing order for labelling the edges. Then it initialises the stacks S1, S2 and S3 to zeros and the initial and terminal nodes are assigned to *v* and *z*. Note that *vprime* is a variable positioning the courant node in the list. Gate is the matrix of results that contains all the MPS and Elapsed time is the duration of the execution.

The execution results are as follows:

```
G =
    0   1   1   0
    0   0   1   1
    0   0   0   1
    0   0   0   0

G =
    0   1   2   0
    0   0   3   4
    0   0   0   5
    0   0   0   0

S1 =   0   0   0   0

S2 =   0   0   0   0

vprime =   1

S1 =   1   0   0   0

S3 =   1   0   0   0   0

S2 =   1   0   0   0

vprime =   2

S1 =   1   1   0   0
```

S3 = 1 0 1 0 0

S2 = 1 3 0 0

vprime = 3

S1 = 1 1 1 0

S3 = 1 0 1 0 1

S2 = 1 3 5 0

S3 = 1 0 1 1 0

S2 = 1 4 0 0

S3 = 1 1 0 0 0

S2 = 2 0 0 0

vprime = 3

S1 = 1 0 1 0

S3 = 1 1 0 0 1

S2 = 2 5 0 0

Gate =

1	3	5	0
1	4	0	0
2	5	0	0

The program generates 3 MPS as showed in the matrix Gate.

5.5 Binary Decision Diagrams

Binary decision diagram (BDD) is a data structure for the symbolic representation of a given Boolean formula and an associated set of decomposition and reduction rules (Lee, (1959)), (Akers (1978)), (Bryant (1986) and (1992)). Boolean formulas are represented using directed acyclic graphs. The mathematical form of a BDD can be written using the decomposition theorem of Shannon or the If-Then-Else function of Bryant. The theorem of Shannon is as follows:

Theorem 5.1 (Shannon)

$F\ x_1, x_2, \cdots, x_i, \cdots, x_n\ =\ x.F\ x_1, x_2, \cdots, 0, \cdots, x_n\ +\ x.F\ x_1, x_2, \cdots, 1, \cdots, x_n$

where *x* is one of the decision variables, and $F_{x=i}$ is the Boolean function evaluated at $x = i$. Based on theorem 5.1, Bryant obtains the ITE function as shown in the following:

5.5.1 ITE function manipulation

Suppose that *A* and *B* are Boolean functions. If *x* and *y* are two variables with an ordering operator (<) on variable, such as, $(x < y)$. Applying now the ITE function for the conjunction and the disjunction operators, we obtain:

$ite\ \ x, A_{x=1}, A_{x=0}\ \ \wedge ite\ \ x, B_{x=1}, B_{x=0}\ \ = ite\ \ x, (A_{x=1} \wedge B_{x=1}\ \ , (A_{x=0} \wedge B_{x=0}))$

$ite\ \ x, A_{x=1}, A_{x=0}\ \ \vee ite\ \ x, B_{x=1}, B_{x=0}\ \ = ite\ \ x, (A_{x=1} \vee B_{x=1}\ \ , (A_{x=0} \vee B_{x=0}))$

$ite\ \ x, A_{x=1}, A_{x=0}\ \ \wedge ite\ \ y, B_{y=1}, B_{y=0}\ \ = ite\ \ x, (A_{x=1} \wedge B\ \ , (A_{x=0} \wedge B))$

$ite\ \ x, A_{x=1}, A_{x=0}\ \ \vee ite\ \ y, B_{y=1}, B_{y=0}\ \ = ite\ \ x, (A_{x=1} \vee B\ \ , (A_{x=0} \vee B))$

and $ite\ \ y, B_{y=1}, B_{y=0}$

By choosing a total order over the variables, and applying recursively Shannon theorem, it is possible to replace the Boolean variable x by a Boolean function g as follows :

$$f_{x=g} = g.f_{x=1} + g'.f_{x=0}$$

Where $f\ and\ g$ are Boolean functions and g' is the complement of g.

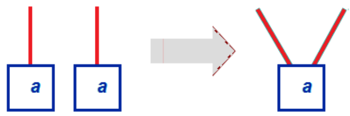
In addition to calculating the recursive function *ite*, we use identities relation to avoid the fact to calculate them again at each time when they occur in a term. These identities are defined as follows:

ite(*f*, 1, 0) = *f* ; *ite*(1, *g* , *h*) = *g* ; *ite*(0, *g*, *h*) = *h* ; *ite*(*f*, *g* , *g*) = *g* ; *ite*(*f*, *0*, 1) = *f'*

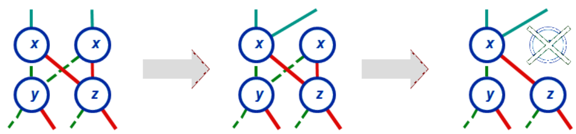
Others basic identities are presented in figure 6 (table in left).

BDD algorithm proceeds by manipulating the binary decision tree using the RP procedure, then the restrict procedure to transform it to a BDD and then to an OBDD (figure 5.5). Finally the ITE (if then else) is applied to get a Reduced BDD. The first decomposition and reduction operations are as follows:

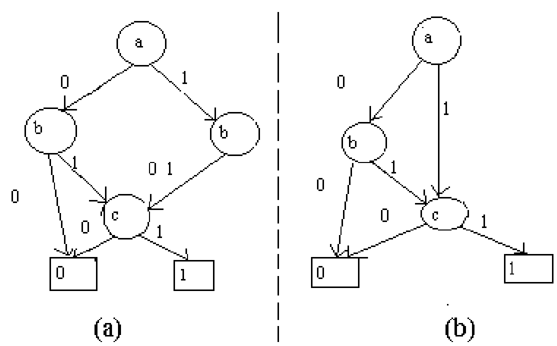
(1) Merging equivalent leaves of a binary decision tree.



(2) Merging isomorphic nodes.



(3) Elimination of redundant tests



The example discussed in the following figure explains the above three steps.

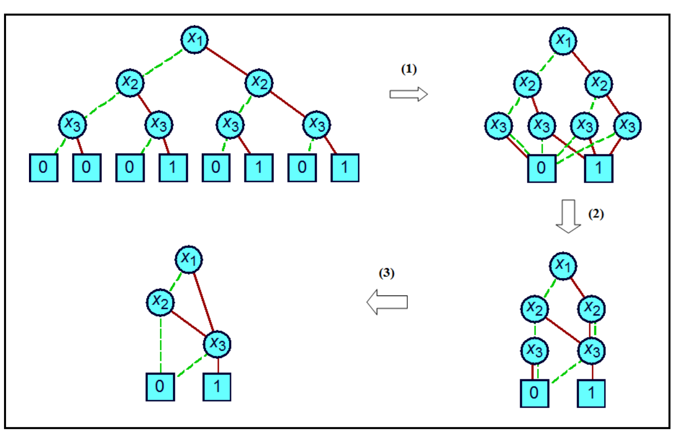


Figure 5.5. Binary Decision Tree and its successive reduction graphs.

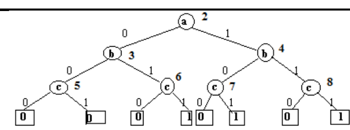
5.5.2 Variable orderings

A BDD is said to be ordered if there is a total ordering of the variables such that every path through the BDD visits nodes according to the ordering. In an ordered BDD (OBDD), each child of a non-terminal node must therefore either be terminal, or non-terminal. The variable orderings is very important to avoid exponential growing of the BDD size. Many publications try to give techniques for simplifying the branches number (Bollig *et al.* (1996).

5.5.3 From OBDD to ROBDD transformation

Any OBDD can be reduced to an ROBDD by repeatedly eliminating in a bottom-up fashion, any instances of duplicate and redundant nodes. If two nodes are duplicates, one of them is removed and all of its incoming pointers are redirected to its duplicate. If a node is redundant, it is removed and all incoming pointers are redirected to its just one child (see Figure 5.2).

Expression	Equivalent form
0	0
1	1
$f.g$	$ite(f,g,0)$
$f.g'$	$ite(f,g',0)$
f	f
$f'g$	$ite(f,0,g)$
$f \oplus g$	$ite(f,g',g)$
$f + g$	$ite(f,1,g)$
$(f+g)'$	$ite(f,0,g')$
f'	$ite(f,0,1)$
$f + g'$	$ite(f,1,g')$
$f' + g$	$ite(f,g,1)$
$(f.g)'$	$ite(f,g',1)$



	0 -	-
	1		
2	a	3	4
3	b	5	6
4	b	7	8
5	c	0	0
6	c	0	1
7	c	1	0
8	c	0	1

Figure 5.6. Left: Table of the simplified Boolean functions presented as an *ite* relation. Right Representation of BDDs in computer.

5.5.4 APPLY Procedure

The APPLY procedure is an efficient tool to combine functions using binary operators like the conjunction and disjunction Boolean operators. It is the major core of our algorithm. It can also be used to complement a function; it suffices in such case to complementing the values of the terminal vertices.

According to Bryant (1992), the APPLY procedure takes graphs representing functions f_1 and f_2, a binary operator (say OP) and produces a reduced graph representing the function $f_1 \;\; OP \;\; f_2$. It proceeds as follows:

$$(f_1 \;\; OP \;\; f_2)(x_1, \cdots, x_n) = f_1\,(x_1, \cdots, x_n) \;\, OP \;\; f_2(x_1, \cdots, x_n)$$

The APPLY procedure also called the compose procedure is based on the recursion derived from theorem 5.1. It could be presented as follow:

$$f_1 \;\; OP \;\; f_2 = x_i.\;\; f_1|_{x_i=0} \;\; OP \;\; f_2|_{x_i=0} \;\; + x_i.\;\; f_1|_{x_i=1} \;\; OP \;\; f_2|_{x_i=1}$$

where x_i is the complement of the Boolean variable x_i.

Let consider two functions f_1 and f_2 representing two Booleans expressions represented by graphs with roots *v*1 and *v*2. Several cases are to be considered. The simplest case is defined by the way when both *v*1 and *v*2 are terminal vertices. Then the resulted graph consists of a terminal vertex having *value* (*v*1) <*op*> *value* (*v*2). Consider now the case where at least one of the two is a non terminal vertex. If *index* (*v*1) = *index* (*v*2) = *i*, we create a vertex *u* having index *i*, and apply the algorithm recursively on *low* (*v*1) and *low* (*v*2) to generate the subgraph whose root becomes *low* (*u*), and on *high* (*v*1) and *high* (*v*2) to generate the subgraph whose root becomes *high* (*u*). Suppose, on the other hand, that *index* (*v*1) = *i*, but either *v*2 is a terminal vertex or *index* (*v*2) > *i*, Then the function represented by the graph with root *v*2 is independent of x_i.. Hence we create a vertex *u* having index *i*, but recursively apply the algorithm on *low* (*v*1) and *v*2 to generate the subgraph whose root becomes *low* (*u*), and on *high* (*v*1) and *v*2 to generate the subgraph whose root becomes *high* (*u*). A similar situation holds when the roles of the two vertices in the previous case are reversed. In general the graph produced by this process will not be reduced. Hence we apply the reduction algorithm before returning the result (for more details read Bryant (Bryan (1992)). The complexity of the APPLY algorithm is exponential in *n,* but by using some refinements this complexity can be reduced to the product of the two graph sizes. This structure of APPLY procedure is presented in Figure 5.7. A similar presented can be found in Bryant (1992). An example showing how APPLY proceeds to compose 2 Booleans expressions is depicted in Figure 8.

```
Procedure APPLY (OP,f1,f2)
     init_cache(C) // cache table be initialised
    APP_STEP(f1,f2)
    If    C(OP,f1,f2) = hit then return C(OP,f1,f2)
      else   if f1 and f2 are terminal then
                 either u = f1 OP f2
              else either u = (if var(f1) < var(f2)  then  place_if_absent T( var(f1),
                                   APP_STEP(low(f1),f2),
                                                      else if var(f1) > var(f2) then
/* idem for the node f2 */
                                 else place_if_absent T( var(f1),
                                   APP_STEP(low(f1),low(f2)),
                                   APP_STEP(high(f1),high(f2))
                                        insert u = (OP,f1,f2) in C
                                        return u
              return APP_STEP(f1,f2)
end.
```

Figure 5.7. Algorithm for composing 2 booleans expressions.

5.5.5 Reliability Evaluation using BDD

From the programming side, using BDDs to compute the reliability is not a difficult way. For that, it suffices to use ITE and APPLY procedures. Generally, the algorithm proceeds as follows :

```
Algorithm Reliability_evaluation (G, v, z)
-     Define the BDD relative to the Boolean 1, call the BDD_ONE
-     Define the BDD relative to the Boolean 0, call the BDD_ZERO
-     Initialise BDD_ALL = BDD_ZERO
-     Apply the procedure ITE to get the BDD of  Boolean TERM, say
BDD_TERM
-     Make BDD_AND = BDD_1
 Do while .true.
 TERM = pathDFS(G, v, z)
   BDD_AND = BDD_AND .and. BDD_TERM
   BDD_OR = BDD_OR .or. BDD_AND
      If last TERM has been encountered
         Return .F.
      else  BDD_AND = BDD_1
      endif
  enddo
end.
```

Figure 5.8. Reliability evaluation algorithm based BDD.

5.5.6 Test example

Suppose after the reduction steps, we get the following ROBDD expression $x_1x_4 + x_1x_3x_5 + x_2x_3x_4 + x_2x_5$. Figure 5.9 shows the initial network (at left) and its ROBDD (at right). Let that the reliability of each component is equal to 0.9. If we apply the algorithm shown the Figure 5.8, its reliability evaluation is computed recursively from the bottom to the top (root) as depicted in the graph 10 (at the right).

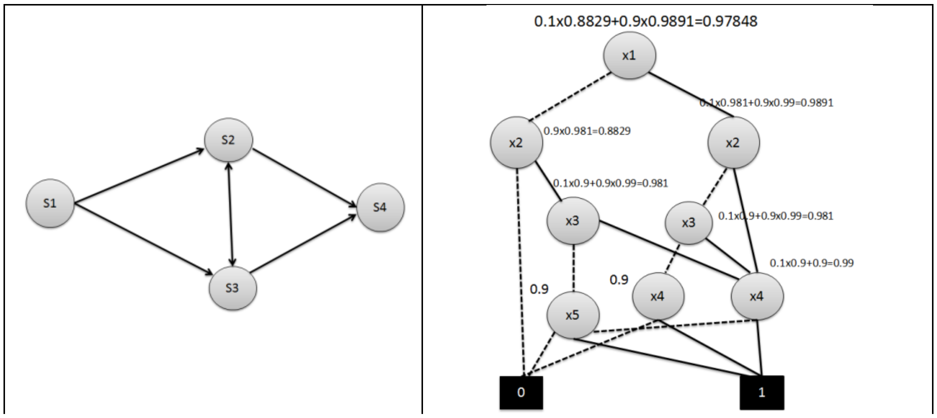


Figure 5.9. A network, its the corresponding BDD and the process for computing the reliability.

5.6. Experimental Results

The proposed algorithm was implemented in Matlab 8 and Java Jdk 1.6. The reference section presents some of them. The program runs on a PC using Vista operating system of Microsoft. We have demonstrated the efficiency of the implementation on some well-known benchmark. There are directed and undirected graphs with variable size (see figure 11 and figure 12) as illustrated in many papers relative to the reliability evaluation. Table 5.1 lists the computational effort providing the number of minimal paths, run time for reliability evaluation and the reliability of each case network. We have compared the performance of the program with the runtime of many others applications and we concluded that for the most cases our algorithm is more efficient.

Table 5.1. Computational results of experimental networks.

Networks	Nodes	Links	Minimal Paths	Runtime Seconds	Reliability value
A	18	29	44	0.060	0.991852559
B	11	18	36	0.031	0.870460090
C	16	30	36	0.95	0.997186290
D	36	60	252	0.368	0.886537674
E	20	60	432	38.062	0.963020072
F	20	60	780	>60	0.99712
G	13	56	1808	1.029	0.989899679
H	17	50	136	>60	0.99806
I	16	48	98	0.063	0.987816744
J	6	20	65	0.117	0.999989960

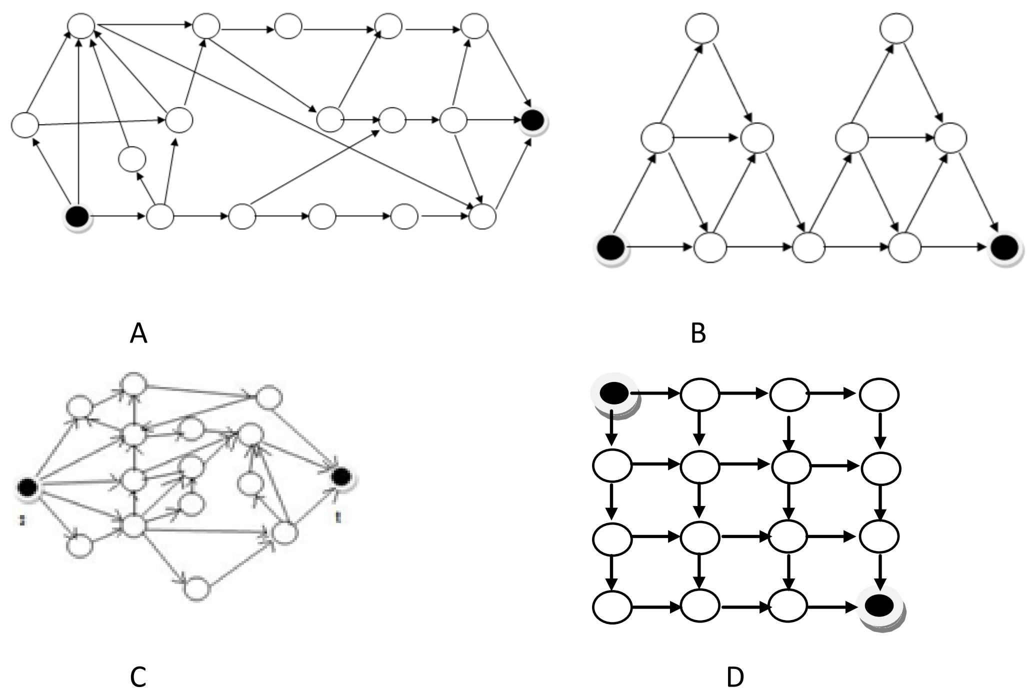
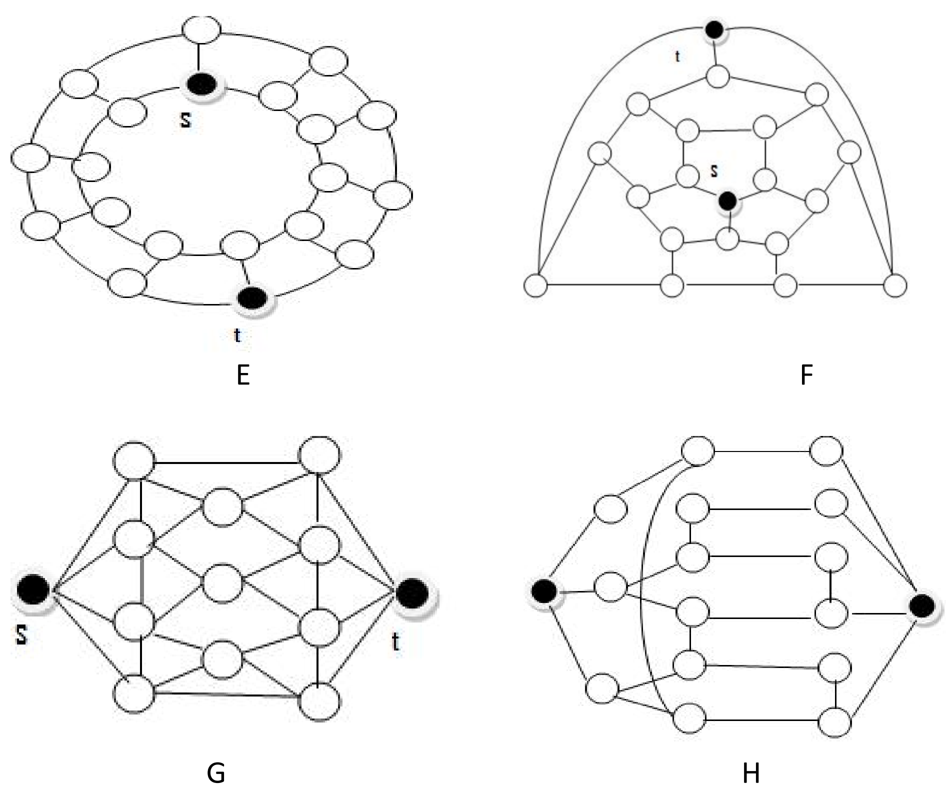
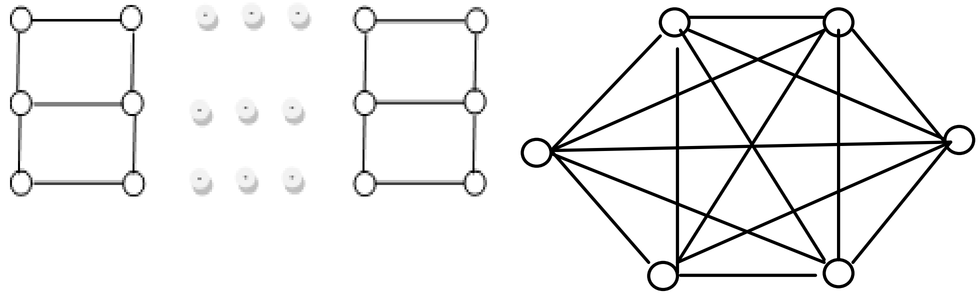


Figure 5.10. A set of directed complex networks





I J

Figure 5.11. A set of undirected complex networks

5.7 CASE STUDY- A RADIO COMMUNICATION NETWORK

To illustrate the performance of the algorithms presented in sections 3 and 4, we propose a practical application to a case study problem of undirected regional radio communication network showed in Figure 5.13. It is radio communication system which is composed of equipments scattered across a wide geographic area. It consists of a set of mobile and portable transmitter-receivers deserved by a, network of fixed equipments located in Quebec province. There are two master site in operation 24 hours a day and a standby third one used in case of urgency, and more than 150 base stations used to transmit the signal generated through the microphone to portable and mobile equipment. A *master site* consists of core and exit routers, WAN and LAN switches, controllers and some operative computers plus others monitoring and dispatching hardware/software systems such as gateway routers, AEB, PBX, dispatching consoles Elite, and so. The radio sites is equipped with one or two antennas for broadband coverage on which is terminated 4 to 8 transmitter-receiver transponders (Tx / Rx). The transponders are connected to each antenna via filtration equipment of type Multicoupler. The multicouplers form a chain of multicoupling able to accept other transponders in expansion. The range of the base station depends on its power, antenna system, terrain, carrier transporter (e.g. T1 or E1) and environmental conditions.

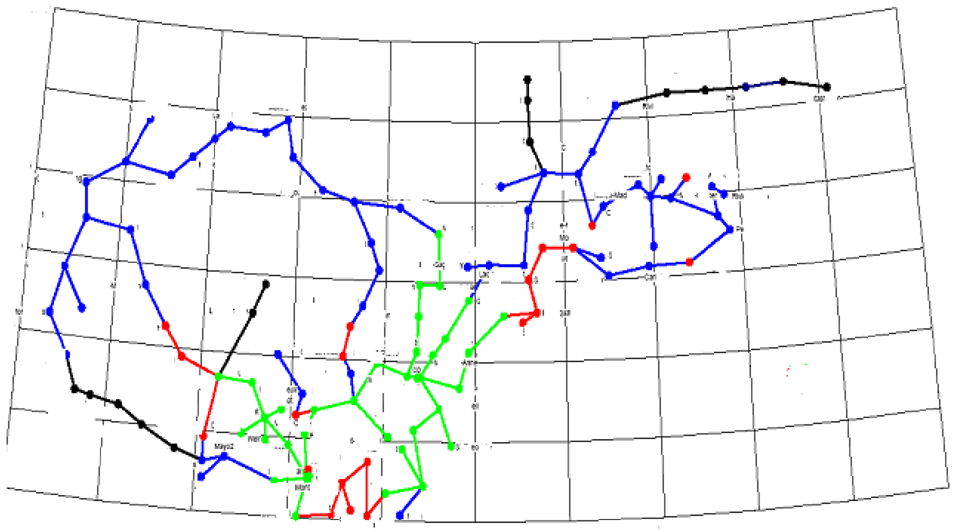


Figure 5.12. A regional radio communication network

The graph in Figure 5.12, presents the radio communication network. Each node is a standard site or a master site. The link between sites from one side is insured using a microwave system and from the other side, between user portables, vehicles and base stations using HF/THF system. The reliability values presented in table 5.2 have been obtained using the calculus system presented in chapter 2. It consists of determining the physical and functional characteristics of each element and from which the reliability is derived using some well-known models (see chapter 2). Please read for example $R_{A,I} = 0{,}9999217518$ which correspond to the value of reliability of the station (A,I). Table 5.3, gives the reliability of the microwave link taken between 2 stations. I is determined using the availability value generated by an electronic system. The reliability in such case is considered equivalent to the availability because the mean time to repair is negligible. The application of the program using the radio-communication system gives the results depicted in table 5.4. The reliability values in table 5.4 correspondent to the 2-terminal reliability (see chapter 1 for details).

Table 5.2. The reliability values on each node of the network

	I	II	III	IV	V	VI
A	0,9999217518	0,9999214639	0,9999226453	0,9999035488	0,9999072594	0,9999676048
B	0,9999312620	0,9998767606	0,9999668616	0,9991865009	0,9998967082	0,9998065989
C	0,9998759277	0,9999355968	0,9999668638	0,9999218563	0,9999215934	0,9999361491
D	0,9999921132	0,9999749248	0,9999522850	0,9998774011	0,9997067831	0,9999356854
E	0,9999814372	0,9999219391	0,9999074941	0,9999670840	0,9999672287	0,9999218934
F	0,999921718	0,999935912	0,9998763	0,99996313	0,9998895	0,9999628
G	0,99992195	0,99992188	0,99995258	0,99986165	0,99987582	0,99987641
H	0,99992253	0,99987673	0,99996689	0,99987692	0,99992182	0,99996659
I	0,99993650	0,99992203	0,99984370	0,99996280	0,99993590	0,99987685
J	0,99993526	0,99990738	0,99996696	0,99986175	0,99992139	0,99996312
K	0,99997443	0,99992150	0,99987681	0,99998115	0,99993612	0,99996659
L	0,99992262	0,99993574	0,99992195	0,99993556	0,99975818	0,99996356
M	0,99992252	0,99987664	0,99992160	0,99988880	0,99986166	0,99990733
N	0,99991806	0,99987706	0,99992142	0,99984402	0,99993629	0,99992187
O	0,99992161	0,99992188	0,99992194	0,99997688	0,99992223	0,99999657
P	0,99997547	0,99996720	0,99937161	0,99996757	0,99987634	0,99999657
Q	0,99945893	0,99980616	0,99990358	0,99980685	0,99992241	0,99999669
R	0,99987688	0,99992187	0,99980593	0,99960116	0,99993537	0,99987610
S	0,99993613	0,99987655	0,99987640	0,99987689	0,99992139	0,99993623
T	0,99992548	0,99996688	0,99990713	0,99986244	0,99988479	0,99998103

Table 5.3. The reliability of each microwave link between two nodes.

	I	II	III	IV	V	VI
A	0,999987	0,999996	0,999998	0,999997	0,999992	0,999994
B	0,999992	0,999999	0,999996	0,999998	0,999999	0,999990
C	0,999986	0,999995	0,999989	0,999998	0,999998	0,999997
D	0,999999	0,999988	0,999985	0,999989	0,999993	0,999990
E	0,999996	0,999986	0,999988	0,999990	0,999982	0,999984
F	0,999970	0,999976	0,999997	0,999982	0,999997	0,999974
G	0,999999	0,999993	0,999998	0,999995	0,999997	0,999988
H	0,999995	0,999991	0,999998	0,999990	0,999999	0,999996
I	0,999999	0,999983	0,999987	0,999995	0,999996	0,999998
J	0,999987	0,999992	0,999990	0,999987	0,999998	0,999982
K	0,999985	0,999995	0,999986	0,999996	0,999997	1
L	0,999994	0,999994	0,999980	0,999993	0,999997	0,999991
M	0,999988	0,999991	0,999984	0,999998	0,999976	0,999998
N	0,999992	0,999992	0,999985	0,999985	0,999974	0,999999
O	0,999996	0,999982	0,999998	0,999999	0,999995	0,999999
P	0,999966	0,999997	0,999996	0,999988	0,999977	0,999979
Q	0,999994	0,999989	0,999999	0,999999	0,999990	0,999989
R	0,999996	0,999997	0,999985	0,999994	0,999997	0,999981
S	0,999998	0,999991	0,999996	0,999977	0,999992	0,999997
T	0,999999	0,999995	0,999999	0,999981	0,999998	0,999998
v	0,999999	0,999999	0,999993	0,999997	0,999987	0,999997

Table 5.4. The reliability joining any two nodes (dimension = 156 x 156 links).

	I	II	III	IV	V	VI
A	0,9998530	0,9997959	0,9999233	0,999843	0,99984429	0,9998582658
B		0,9998054	0,9997880	0,999853	0,99985380	0,9998677754
C			0,9999312	0,999717	0,99971846	0,9997324326
D				0,999914	0,99991464	0,9999286228
E				0,999697	0,99969812	0,9997120964
F				0,999841	0,99984247	0,9998564536
G					0,99984449	0,9998584678
H						0,9998590481

5.8 CONCLUSION

A method for evaluating the 2-terminal reliability has been proposed in this paper. We have used depth first search (DFS) algorithm for minimal paths set discovering and Binary Decision Diagrams (BDD) for reduction and evaluation of networks reliability. The program runs on some well-known benchmarks and gives good execution time. We remain convinced that the program will operate on more complex instances of hundreds of nodes and links and even more. We have applied the program to a regional radio communication network. Despite that the evaluation of the reliability is a big problem in our case, the program have computed the reliability of any path taken between any two nodes of the network and finally by composing all these reliabilities we compute the reliability of the network. We confirm that all the computing procedures are executed in a finite time not exceeding some seconds.

REFERENCES

Abraham J. A., 1979, An Improved Algorithm for Network Reliability, *IEEE Transactions on Reliability*, pp. 58–61.

Akers. S. B., 1978, Binary Decision Diagrams, *IEEE Transactions on Computers*, Vol. C-27, No 6, pp. 509–516.

Andersen H. R., 1999, An Introduction to Binary Decision Diagrams, *Lecture Notes, IT University of Copenhagen*.

Bollig B., Ingo W., 1996, Improving the Variable Ordering of OBDDs Is NP-Complete, *IEEE Transactions on Computers*, Vol. 45, No 9, pp 993–1002.

Bryant. R. E., 1986, Graph-Based Algorithms for Boolean Function Manipulation, *IEEE Transactions on Computers*, Vol. C-35, No 8, pp. 677–691.

Bryant. R. E., 1992, Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams, *ACM Computing Surveys*, Vol. 24, No. 3 , pp. 293–318.

Coudert O. and J. C. Madre, 1992, Implicit and incremental computation of primes and essential primes of Boolean functions, in *Proc. of the 29th ACM/IEEE Design Automation Conference*, *IEEE, Computer Society Press*, pp. 36–39.

Dotson W. P., J. 0. Gobien., 1979, A new analysis technique for probabilistic graphs, *IEEE Trans*, Circuits and Systems, Vol. Cas-26, NO. 10, pp- 855-865.

Hardy. G., C. Lucet, and N. Limnios., 2007, K-Terminal Network Reliability Measures With Binary Decision Diagrams, *IEEE Transactions on Reliability*, Vol. 56, NO. 3, pp. 506-515.

.

Heidtmann K.D., 1991, Smaller Sums of Disjoint Products by Subproduct Inversion, *IEEE Transactions on Reliability*, Vol. 38, pp. 305-311.

Institute of Electrical and Electronics Engineers, 1990, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York, NY.

Jasmon, G. B. & O. S. Kai., 1985, A New Technique in Minimal Path and Cutset Evaluation, *IEEE Transactions on Reliability*, Vol. R-34, no. 2, pp. 136-143.

Lee. C. Y., 1959, Representation of Switching Circuits by Binary-Decision Programs, *Bell Systems Technical Journal*, Vol. 38, pp. 985–999.

Lin, H. Y., Kuo, S. Y. and Yeh, F. M, 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, in *Proc. of the Eighth IEEE International Symposium on Computers and Communication*, Vol. 2, pp. 1341-1346.

Liu, H. H., Yang, W. T. and Liu, C. C, 1993, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronic Reliability*, Vol. 33, NO 4, pp. 599-613.

Locks, M. O. and Wilson, J. M, 1992, Note on disjoint products algorithms, *IEEE Transactions on Reliability*, Vol. 41, NO. 1, pp. 81-84.

Rudell, R, 1993, Dynamic variable ordering for ordered binary decision diagrams, in *Proc. of the 1993 IEEE/ACM international conference on Computer-aided design*, pp. 42-47.

Shier D. R., D.E. Whited, 1985, Algorithms for generating minimal cutsets by inversion, *IEEE Transactions on Reliability*, Vol R-34, pp 314- 319.

Tarjan, R, 1972, Depth-first search and linear graph algorithms, *SIAM Journal of Computing*, Vol. 1, NO. 2, pp. 146-160.

Theologou R., J.G. Carlier, 1991, Factoring & Reduction for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 40, NO. 2, pp. 210-217.

Chapter 6

A Practical Algorithm for Network Reliability Evaluation Based on the Factoring Theorem – A Case Study of a Generic Radio communication System

6.1 ABSTRACT

The evaluation of complex systems reliability is a crucial test to secure operations and infrastructure against failures. This paper presents the network decomposition method and compares some results to solve network reliability problem. The algorithm proceeds recursively according to the factoring theorem in conjunction with simplification and polygon-to-chain reduction rules and considers the case where network components-vertices and edges could fail randomly. The implementation of the algorithm is efficient and proceeds in less time comparing with the best examples found in the literature.

6.2 INTRODUCTION

Checking the reliability of distributed finite-state systems is a big challenge due to the state explosion problem, which occurs in practice for realistic systems with complex data and structure. Network reliability has received full attention from designers for the validation of systems like those of telecommunication, electric and water networks. The IEEE 90 standard (IEEE, 1990) defines the reliability as the ability of a system to perform its required functions under stated conditions for a specified period of time. In the most case, system network is modeled using a particular data structure defined as a graph. Generally, the graph components are vertices (nodes) and arcs (edges), and a chain P in a graph is an alternating sequence of distinct vertices and edges, such that the internal vertices are of degree greater than 2. A cut C is a minimal set of edges whose removal breaks all directed paths between the source **s** and the sink **t**. A chain P and a cut C are minimal if they have no proper sub-paths and sub-cuts. If two chains P_1 and P_2 have common vertices u and v, ie, the chains are parallel, then $P_1 \cup P_2$ is a polygon. Two chains are series if they have a common vertex. A *k-tree* of a graph G_k is any minimal sub-graph that connects all the K-vertices of G_k. G_k is connected if its k-vertices are connected and a formation of a k-graph is a set of k-trees of the graph whose union yields the graph. The problem of finding all minimal sets (minpaths and mincuts) could be enumerated using the Tarjan's depth first search procedure (for minimal paths) (Tarjan, 1972) or the algorithm of Yan and Taha (for minimal cutsets) (Yan, *et al*., 1994). For example, Lin *et al.* (2002) have used an interesting algorithm showing that a 2×100 lattice network (Figure 4) has 2^{99} paths but contains 10,000 minimal cutsets.

Technically, there are two main families of algorithms called: (1) SDP for sum of disjoints products (Abraham, 1979; Fratta and Montanari, 1973; Heidtmann, 1989; Liu, *et al.*, 1993; Locks and Wilson, 1992), and (2) those based on the well-known factoring problem (Bryant, 1986; Theolougou and Carlier, 1991; Wood, 1986). In the former techniques, Abraham in (Abraham, 1979), following the work of Frotta and Montanari (Fratta and Montanari, 1973) determines first, minmal paths between a pair of nodes (e.g source and sink), where paths terms correspond to a sum of disjoint products (mutually exclusive) from which the reliability expression can be directly computed. Several authors have given different procedures and the aim has always been to reduce the number of terms in the SDP (Liu, *et al.*, 1993). Despite these improvements in case of complex networks, the problem becomes drastically unsolved and the needs of methods and formalisms to diminish the size of Boolean product terms still a welcome way. It is known in the practical cases that the network reliability algorithms belong to the class of NP-hard problems, and in the theory they have been classified as #P-complete by Valiant (Valian, 1979). Generally, Boolean algebra and probability theory are used to modify additive minsets to an equivalent set of mutually exclusive minsets (Fratta and Montanari, 1973). Heidtmann gives in (Heidtmann, 1989) an elegant algorithm which reduces considerably the size of formula. Among the later techniques based on the factoring theorem, the algorithm due to Dotson and Gobien (1979) is considered by Yoo *et al.* (Resende, 1986) as the most efficient one among four tested algorithms. Wood has presented in (Wood, 1986) some elegant methods that reduce the graph using eight rules, called: parallel reduction, series reduction, degree-2 reduction, polygon-to-chain reduction, bridge contraction, irrelevant component deletion, degree-3 reduction and trivial reduction. Theologou and Carlier (1991) give an interesting heuristic which consider the case of imperfect nodes (nodes subject to failure) without replacing each imperfect node by two perfect ones and adding between them an arc with reliability equal to that of the node. In Yoo and Deo (1988), the authors compare four algorithms for the terminal-pair reliability problem and show that the algorithm due to Dotson and Gobien is the most efficient, which algorithm is presented in Dotson and Gobien (1979). Other solutions have been found using BDDs (binary Decision Diagrams) (Brace, *et al*., 1990; Bryant, 1986), and their extended ROBDD (Reduced Ordered BDD) (Rudell, 1993). It has been shown in (Hardy, *et* al., 2007; Kuo, *et al.*, 2007; Lin, *et al*., 2003), that reliability can be computed by constructing the symbolic reliability function with BBD.

This paper presents an algorithm that guarantees the evaluation of networks reliability in case of imperfect nodes without computing minsets. It is based on the work of Carl Simard (Simard, 1996) and Wood *et al.* (1986). This operation has reduced considerably the size of benchmark networks. Thus, a series of tests and comparisons have been applied using some well-known benchmarks. The algorithms developed in this paper are efficient and take less time than others demonstrations.

The organization of the paper is as follows. In the next section readers are briefed on preliminaries. Section 3, presents the factoring theorem and its extension to cover network reliability class of problems. In Section 4, the derivation of reduction rules is established. We detail algorithms, tests and comparisons in Sections 5 and 6. Section 7 introduces a radio-communication system and its reliability evaluation. Finally, concluding remarks are given in Section 8.

6.3 Preliminaries

A network is a graph $G = (V, E)$ where V is a finite set of vertex (nodes) and E is a finite set of edges or arcs for directed graphs. Each edge and each node of the graph can be weighted with a real value p_i corresponding to the probability that a component (the edge) does not fail (is good), and called reliability. The nodes are numbered from *1* to n, and edges from *1* to m. We denote the K-terminal reliability of G by $R_k(G)$, where K is a specified subset of V with $|K| > 2$ (2: means one source and one sink). In the most cases, it suffices to have a 2-terminal relation to compute the reliability evaluation. A success set, is a minimal set of the edges of G such that the vertices in K are connected; the set is minimal so that deletion of any edges causes the vertices in K to be disconnected. Topologically, a success set is a minimal tree of G covering all vertices in K. It is defined that parallel edges are edges with the same *end* vertices and non parallel edges are adjacent if they are incident on a common node. Two adjacent edges are series edges if their common node is of degree 2 and is not in

K. It is noted that replacing a pair of series (parallel) edges by a single edge is called series (parallel) reduction. So, if e is an edge with *end* vertices u and v, then $(G-e)$ is a sub-graph of G obtained by deleting e from G and $(G*e)$ is a sub-graph of G obtained by contracting edge e from G. Edge e is known as the keystone edge, which cannot be chosen arbitrarily.

6.4 THE PRINCIPLE OF FACTORING DECOMPOSITION

Network factoring theorem is the basis for a class of algorithms for computing K-terminal reliability (Page and Perry, 1988; Bryant, 1986; Wood, 1986; Choi and Jun, 1995; Deo and Medidi, 1992; Hardy, *et al*., 2007; Page and Perry, 1989). The topological interpretation of the simple conditional reliability formula for a general binary system ***S*** with component e_i is,

$$R(S) = p_i R(S|e_i works) + q_i R(S|e_i fails) \tag{6.1}$$

where $R(S)$ is the reliability of the system S and $R(S|e_i works)$ is the reliability of the system S when the edge e_i is in operation and $R(S|e_i fails)$ is the reliability of the system S when the edge e_i is not in operation and each probability p_i is asserted to the edge i and $q_i = (1-p_i)$ the opposite of p_i. Figure 6.1 details the application of the factoring decomposition process on the bridge network.

Techniques using factoring theorem are well-suited for evaluating network reliability because they can be combined with other techniques. The association of different techniques depends on some conditional constraints. For example, in case of undirected graphs, an edge is replaced by two anti-parallel directed edges (arcs), having the same reliability but functioning statistically independent from each others. More problems arise in such transformation which needs more attention than for the undirected networks.

According to the following development, we assume that a network could be (1) directed or undirected, (2) the edges may be in operation or not, (3) in case of failure, vertices and edges are statistically independents and their probabilities are known, (4) the problem of estimating the network reliability means to determine the existence of a path linking the node ***s*** (sink) to the node ***t*** (target), or to transform the network by a successive reductions until an irreducible form is reached, which could be used to compute its reliability. An imperfect network is defined in the case that at least the probability assigned to a vertex is different from the value 1. In the opposite, the network is perfect when the probability assigned to each node is equal to 1.

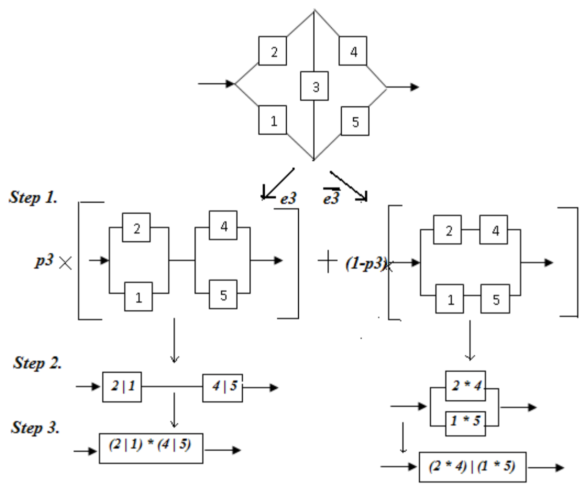


Figure 6.1. A demonstration of factoring series-parallel reductions

Thus, we summarize that the factoring theorem establishes the validity of the following conditional reliability relation:

$$R(G) = p_e R(G * e) + (1 - p_e) R(G - e) \tag{6.2}$$

where G is a graph with edge e labelled by the probability p_e, and $(G-e)$ and $(G*e)$ as defined in preliminaries. Factoring consists of picking an edge of the graph and decomposing the original problem with respect to the possible state of the edge.

6.5. IMPLEMENTING NETWORK RELIABILITY CALCULUS

6.5.1. Decomposition strategy

To be more precise in the case of an undirected network, the solution is given by decomposing each edge into two opposite directed arcs, which are labelled with the same reliability value. Thus, both arcs maintain the same status (success or fail). The choice of a global decomposition strategy can improve the complexity of a factoring algorithm. Such concept has been used by many authors as in (Kuo, *et al*., 2007; Theolougou and Carlier, 1991). As we consider that the nodes could fail, the decomposition strategy proceeds on the edge and its extremities to avoid the complexity. The problem in such case is to find a complete algorithm which could reproduce exactly the form of graph structure, as in (Choi and Jun, 1995; Wood, 1986). To deal with the problem we have written a procedure using the incoming and outgoing degrees corresponding to each node according to the following definitions.

6.5.2. Reduction rules and partitions transformation

In order to reduce the size of the network which leads to diminish the computing cost of the network reliability, we need to apply some reduction techniques. The idea behind the reduction is to transform each graph partition into a simplified form, while preserving its reliability. They are (reductions) similar to those of the factoring theorem, which consist of the replacement of a particular structure (e.g. a polygon) embedded in the graph within the abstraction of the rest of the graph. To demonstrate such procedure we introduce the following reduction rules:

1. Let $e_a = (u, v)$ and $e_b = (u, v)$ be two parallel edges in G_K (the network graph) and suppose that $p_i = 1 - q_i$ ($i = a$ or b). A parallel reduction obtains G' by replacing e_a and e_b with single edge $e_c = (u, v)$ with reliability $p_c = (1 - q_a q_b)$, and it defines $\Omega = 1$ and $K' = K$. We note that Ω is a multiplicative operator derived from $R(G_K) = \Omega.R(G'_{K'})$

2. Let $e_a = (u, v)$ and $e_b = (u, w)$ be two series edges in G_K such that degree(v) = 2 and $v \notin K$. Applying reduction procedure leads to the sub-graph G' by replacing e_a and e_b with a single edge $e_c = (u, w)$ and the corresponding reliability is computed by $p_c = p_a p_b$, and it defines $\Omega = 1$ and $K' = K$.

3. Let $e_a = (u, v)$ and $e_b = (u, w)$ be two series edges in G_K such that degree(v) = 2 and $u, v, w \in K$. A degree-two reduction obtains G' by replacing e_a and e_b with single edge $e_c = (u, w)$ with reliability $v \notin K$ $p_a = \dfrac{p_a p_b}{1 - q_a q_b}$, and it defines Ω = 1 and $K' = K$.

4. Two parallel chains between two vertices are replaced by a new chain. Such operation is called polygon-to-chain reduction.

To demonstrate the results of these transformation steps, let us consider the case of the reduction polygon-to-chain of type1 with imperfect nodes (Figure 2-- presents a simple polygon-to-chain reduction). For that, we suppose that nodes *a* and *b* have status "good" and their respective reliabilities are p_a and p_b, and consider the edges e_1, e_2 and e_3 linking the nodes *s*, *a* and *b*. The application of the factoring theorem using the link (S, e_1, a) gives the reduced subgraphs $(G - e_1)$ and $(G * e_1)$ (as it is shown in Figure 6.2). Thus, the reliability of the graph *G* is given by equation (3),

$$R(G) = p_1 p_a R(G * e_1) + (1 - p_1 p_a) R(G - e_1) \tag{6.3}$$

and the symbolic expression of the reliability of the node *a* is given by (6.4).

$$p_a^{'} = \frac{p_a q_1}{q_a + p_a q_1} \tag{6.4}$$

The following figures demonstrate the successive transformations (step 1) of the factoring theorem (see Figure 6.2).

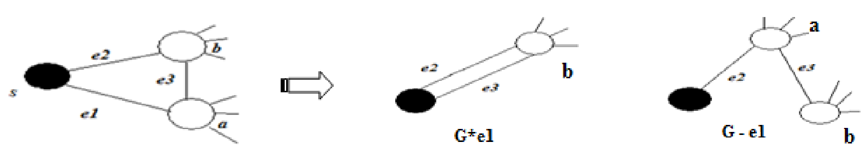


Figure 6.2. Graphs induced by the decomposition on the link e_1

We continue the decomposing of the graph using the link (*S, e_2, b*). The new expression of the reliability is given by equation (6.5).

$$R(G) = p_1 p_a \left[p_2 p_b R((G*e_1)*e_2 + (1-p_2 p_b) R((G*e_1)-e_2) \right] + (1-p_1 p_a)\left[p_2 p_b R((G-e_1)-e_2) \right] \tag{6.5}$$

and the reliability corresponding to the node *b* changes to :

$$p_b^{'} = \frac{p_b q_2}{q_b + p_b q_2} \tag{6.6}$$

Now, using the link (*S, e_3, b*), the algorithm generates the graphs $(G*e_1)-e_2$ and $(G-e_1)*e_2$.

Finally after that, the reliability of the graph is evaluated by the following expression:

$$\begin{aligned} R(G) = p_1 p_a &\left[p_2 p_b R((G*e_1)*e_2) + (1-p_2 p_b) \begin{bmatrix} p_3 p'_b R(((G*e_1)-e_2)*e_3) + \\ (1-p_3 p'_b) R(((G*e_1)-e_2)-e_3) \end{bmatrix} \right] + \\ &(1-p_1 p_a)\left[p_2 p_b \left[p_3 p'_a R(((G-e_1)*e_2)*e_3) + (1-p_3 p'_a) R(((G-e_1)*e_2)-e_3) \right] \right] \end{aligned} \tag{6.7}$$

From this result, we deduce the new reliabilities corresponding to the nodes *a* and *b* as follows:

$$p_a^{''} = \frac{p_a q_1 q_3}{q_a + p_a q_1 q_3} \tag{6.8}$$

$$p_b^{''} = \frac{p_b q_2 q_3}{q_b + p_b q_2 q_3} \tag{6.9}$$

After arranging and suppressing identical expressions, the equation (6.7) is reduced to (6.10).

$$R(G) = p_a p_b \left[p_1 p_2 + p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3\right] R((G * e_1) * e_2)$$
$$+ p_1 p_a \left[1 - p_2 p_b - p_3 p_b + p_2 p_3 p_b\right] R(((G * e_1) - e_2) - e_3) \tag{6.10}$$
$$+ p_2 p_b \left[1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a\right] R(((G - e_1) * e_2) - e_3)$$

Now, the expression of the reliability relative to the graph *G'* is obtained by transforming *G* in which two new edges $e_1^{'}$ and $e_2^{'}$ have been created (see Figure 6.3). Such reliability is expressed by:

$$R(G') = p'_1\, p'_a . R(G' * e'_1) + (1 - p'_1\, p'_a) . R(G' - e'_1) \tag{6.11}$$

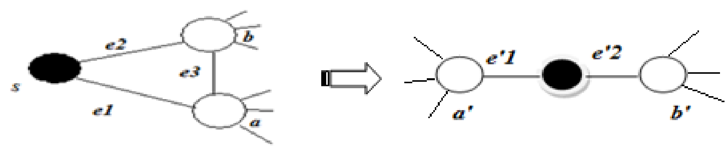


Figure 6.3. A polygon-to-chain reduction of type 1

and the reliability expression of the node *a* in the graph $(G' - e'_1)$ is :

$$(p_a^{'})^{'} = \frac{p_a^{'} q_1^{'}}{q_a^{'} + p_a^{'} q_1^{'}} \tag{6.12}$$

Now, we identify the reliability of the graph *G'* using the expression (6.13),

$$R(G') = p'_1\, p'_a \left[p'_2\, p'_b\, R((G' * e'_1) * e'_2) + (1 - p'_2\, p'_b) R((G' * e'_1) - e'_2)\right]$$
$$+ (1 - p'_1\, p'_a)\left[p'_2\, p'_b\, R((G' - e'_1) * e'_2)\right] \tag{6.13}$$

and, the new expression of the reliability of the node *b* becomes:

$$(p_b^{'})^{'} = \frac{p_b^{'} q_2^{'}}{q_b^{'} + p_b^{'} q_2^{'}} \tag{6.14}$$

As the graph *G* and *G'* are identical, thus we can write that :

$$R[(G * e_1) * e_2] = R[(G^{'} * e_1^{'}) * e_2^{'}] \text{ and } R[(G * e_1) - e_2) - e_3] = R[(G^{'} * e_1^{'}) - e_2^{'}]$$

The last expression is valid if and only if

$$p^{''}_a = (p_a^{'})^{'}\ ;\ R[(G - e_1) * e_2) - e_3] = R[(G^{'} - e_1^{'}) * e_2^{'}] \text{ and } p^{''}_b = (p_b^{'})^{'}.$$

Finally from the equivalence of the relation $R(G) = \Omega.R(G^{'})$, and from the last equalities, we deduce the following system of equations :

$$\begin{cases} p_a p_b [p_1 p_2 + p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3] = \Omega [p_1^{'} p_2^{'} p_a^{'} p_b^{'}] \\ p_1 p_a [1 - p_2 p_b - p_3 p_b + p_2 p_3 p_b] = \Omega [p_1^{'} p_a^{'} (1 - p_2^{'} p_b^{'})] \\ p_2 p_b [1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a] = \Omega [p_2^{'} p_b^{'} (1 - p_1^{'} p_a^{'})] \\ \dfrac{p_a q_1 q_3}{(q_a + p_a q_1 q_3)} = \dfrac{p_a^{'} q_1^{'}}{(q_a^{'} + p_a^{'} q_1^{'})} \\ \dfrac{p_b q_2 q_3}{(q_b + p_b q_2 q_3)} = \dfrac{p_b^{'} q_2^{'}}{(q_b^{'} + p_b^{'} q_2^{'})} \end{cases} \tag{6.15}$$

After solving the system (6.15) for the five unknown parameters $\Omega, p_1^{'}, p_2^{'}, p_a^{'}, p_b^{'}$, we obtain the expressions of δ, A, B, C and D according to (6.17) :

$$\begin{aligned} &\Omega = \frac{(\delta + A)(\delta + B)}{\delta}; \\ &p_1^{'} = \frac{(\delta)}{(\delta + AC)}; \\ &p_2^{'} = \frac{(\delta)}{(\delta + B.D)}; \\ &p_a^{'} = \frac{(\delta + AC)}{(\delta + A)}; \end{aligned} \tag{6.16}$$

$$p'_b = \frac{(\delta + B.D)}{(\delta + B)}$$

$$\left\{\begin{array}{l} \delta = p_a p_b \left[p_1 p_2 + p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3 \right] \\ A = p_2 p_b \left[1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a \right] \\ B = p_1 p_a \left[1 - p_2 p_b - p_3 p_b + p_2 p_3 p_b \right] \\ C = \dfrac{p_a q_1 q_3}{(q_a + p_a q_1 q_3)}; \quad D = \dfrac{p_b q_2 q_3}{(q_b + p_b q_2 q_3)} \end{array}\right. \tag{6.17}$$

Finally, we note that we can derive successive reduction steps to transform a complex topology embedded in a network to a simple edge relaying two nodes. For example concerning the graph structure of the Figure 6.2, the last transformation is shown in the Figure 6.3. We note again that the problem in the case of the factoring theorem is to find an automatic way to recognize the topologies (polygons). The idea is much fruitful and need to be worked well. In our case, we have developed an algebraic way to derive formulae according to each transformation topology. These results have been included in a whole algorithm giving effective results which could be compared with those of many authors.

6.6 ALGORITHMS

To compute the reliability of a network, the algorithm proceeds by successive steps using an iterative procedure (*Cleaning(.)*) to avoid all particular situations which could stop the computation (e.g. false sources or false sinks). After that, a reduction function is called to eliminate redundancy parts using simple reduction rules (*Reduction (.)*). The main process is accomplished by looking for complex sub-structures and if one is found, the program proceeds recursively by separating the structure into two parts using the factoring implementation and continue to do so until no reduction occurs. At the end, the reliability is computed on the fly according to the following algorithm.

```
Algorithm Facto(Adj(G))  /** Adj(G) : adgacency matrix of the network G  **/
Omega = 1; Bool = 0 ;
Do while .T.
```

```
  Do while no-reduction can be applied do
    begin
      Cleaning(G);/ *** remove edges incoming to the node source or outgoing from the
              sink ***/
      Reduction(G); /apply the reduction rules whenever it is possible.
    enddo;
      if s = t then return (Omega) endif
      if out-Degree(s) = 0  .or.  in-Degree(t) = 0 then return(0)
          else  if s and t are connected,
              bool =  1    /*** there is a path between s and t ***/
          else return(0)  endif
     endif
      if bool = 1
        KeyStoneChoise() /*** picking an edge e as defined  ***/
      endif
Construct G1 = G-e_i and G2 = G*e_i ;
        R = Omega. (p_ei.R(G1) + (1- p_ei).R(G2))
 endwhile
endwhile
Return (R)
end (Facto)

Function Cleaning(G)
 Rule 1: All edges going into the source and all edges going out of the sink are removed;
 Rule 2: Every node, except the source and the sink, with 0 in-degree or 0 out-of-degree can
be removed
end(Cleaning)

Function Reduction(G)
    Do case
```

```
      case type 1   do   reduction type-1
                  ⋮
      case type i   do    reduction type-i
    endcase
end (Reduction)

/****          Body of main program      ****/
 read the network G from data file
 Facto(Adj(G),R, s, t)
end.
```

6.7 IMPLEMENTATION AND COMPARISON

We compared the performance of the program written out in MatLab code according to the logic of the algorithm developed in this work and supporting all the factoring phases with some best implementations found in the literature. The program runs on a Personal computer under Microsoft Windows XP operating system. We have applied a procedure which uses respectively simple reductions and polygon-to-chain reductions and tested them by using the benchmarks of the Figures 6.4 and 6.6. The performance of the procedure is presented in the Figure 6.5 and the number of reduction operations in the Table 6.1. We noticed that the efficiency of such reductions are executed faster and diminish the number of decompositions required to compute the reliability. The evaluation of the reliability of each network is given for perfect and imperfect nodes, and this, is based on an initial probability value equal to 0.9 assigned to the imperfect nodes and to each edge. We note that the execution times are very interesting compared with others implementations such as those of the paper (Choi and Jun, 1995) (see Table 6.1).

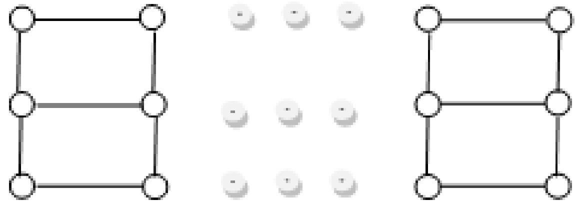


Figure 6.4. Topology of the (3 x m) networks.

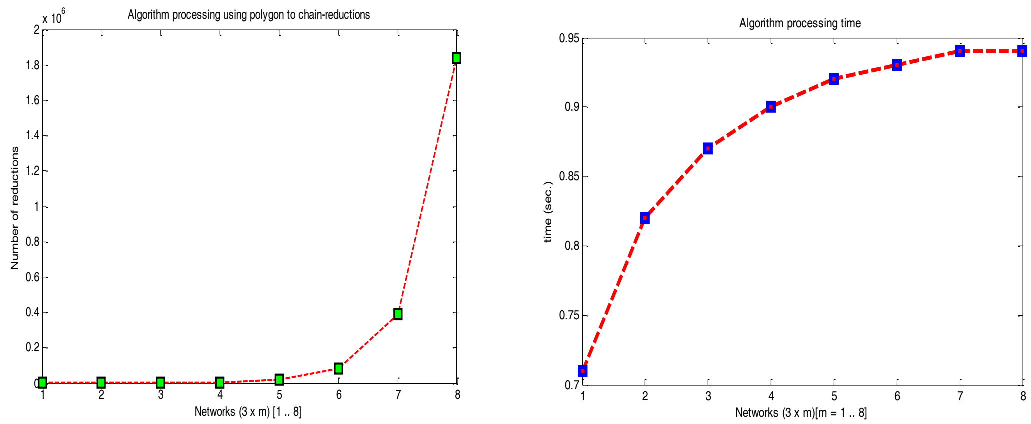


Figure 6.5. Performance of the algorithm using the (3 x m) network

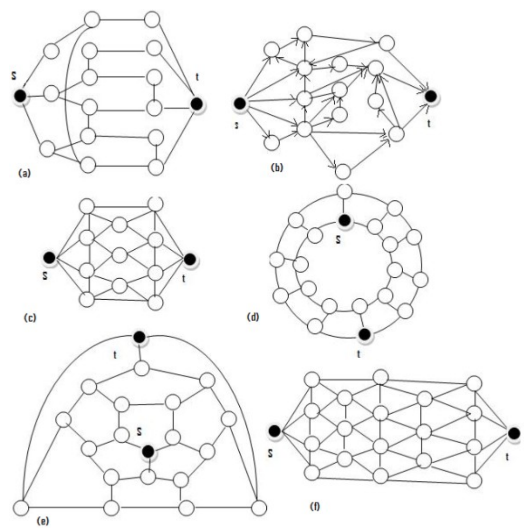


Figure 6.6. A set of networks used as a benchmark.

Table 6.1. Comparison between Choi and Jun (1995) and our algorithm performance.

Reference/Networks	(a)	(b)	(c)	(d)	(e)	(f)
Choi and Jun, 1995	473	???	2620	???	6286	1484937
0ur procedure	318	29	1131	618	4446	370139
Reliability value	0.998059	0.997186	0.99979	0.963020	0.99712	0.999795

Note that the values in Table 6.1 represent the number of the reduction operations for each network and “???” means that the relative network have not been considered by Choi and Jun.

6.8. MODELING A RADIO COMMUNICATION

A conceptual design for a generic radio system provides a convenient mean for people to communicate instantaneously engaged in various public safety-related services.

The generic radio system is usually expected to transmit and receive to its coverage areas radio signals that are used to carry voice and data on a daily basis throughout the year and whatever the critical conditions. An example of a radio communication system is depicted in Figure 7.

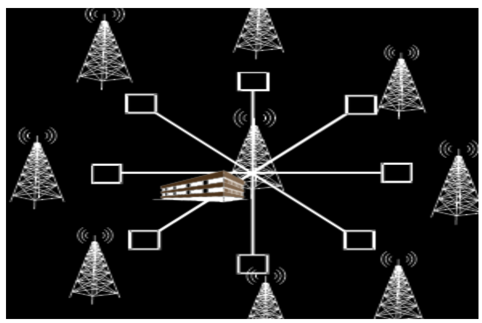


Figure 6.7. A star network topology

A radio system consists of equipment scattered across a wide geographic area. Radio equipment can be classified as either fixed, mobile, or portable and includes at least a transmitter, a receiver, and antenna system (e.g. in Figure 6.8). Fixed equipment is located at a central site such as a headquarters (Figure 6.7), and usually consists (at least) of a base station, microphone, and antenna. The base station is used to transmit the signal generated through the microphone to portable and mobile equipment in a wide area deserved by the system. The range of the base station depends on its power, antenna system, terrain, carrier transporter (e.g. T1 or E1) and environmental conditions.

In a radio infrastructure, system may perform three types of communication: simplex, half-duplex, and duplex. The communication type used depends on the number of users and the type of equipment available.

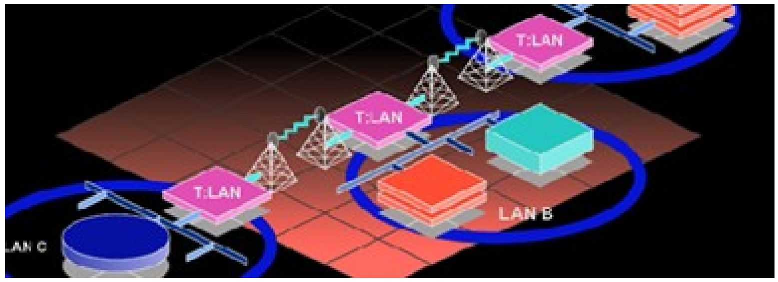


Figure 6.8. A linear network topology

The basic components of the radio system are interconnected forming a network. Such network may contain the following elements:

- **Radios** (portable and mobile radios)
- **Sites** (master sites, secondary Radio Frequency (RF) sites)
- **Zones** (a zone or zones containing one or more RF sites)
- **System** (single zone or multiple zones containing one or more RF sites)

In the radio system, a zone is responsible for managing its own elements (sites, repeaters, subscribers, UHF-VHF and microwave carriers) interconnected using a Local Area Network (LAN) (Figures 6.8 and 6.9). The LANs are interconnected though a high-speed transport network to form a Wide Area Network (WAN). The WAN allows user configuration information, call processing information, and audio to be conveyed throughout the system.

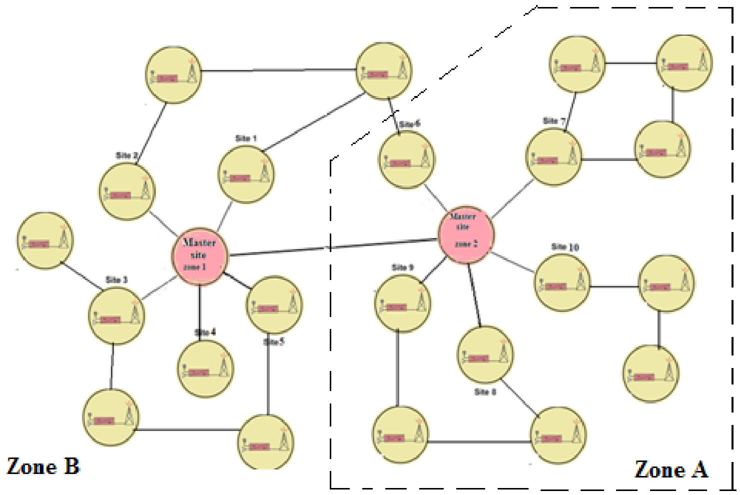


Figure 6.9. A communicating radio network

Each zone framework includes a physical infrastructure, managing mobility and processing calls transported using IP (Internet Protocol) packet technology through the network.

An *ordinary radio* site consists of the following subsystems (see Figure 6.10):

- A tower
- UHF-VHF antennas
- Microwave antennas
- Connectors
- Power electric energy Resources deserved by a public network and batteries used to secure power alimentation in case that the power network fail to deliver energy.
- Repeaters resources stations (communicating channels)

A *master site* consists of core and exit routers, WAN and LAN switches, controllers and some operative computers plus others monitoring and dispatching hardware/software systems.

The generic radio communication system is developed to support the development of an analysis methodology for Reliability, Availability and Maintainability (RAM) and thus to optimize the performance of the system.

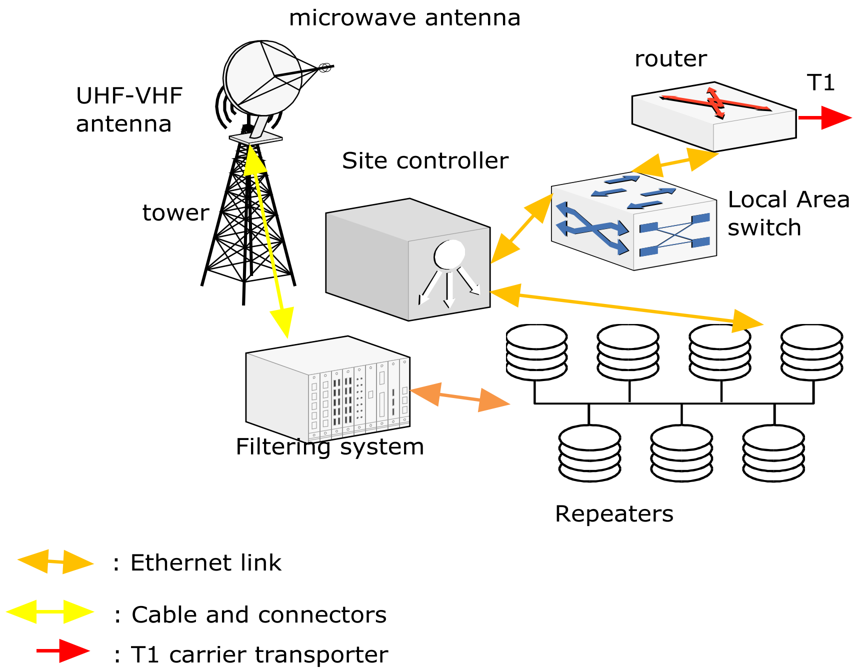


Figure 6.10. A standard radio communication site

Based on the RAM requirement it is expected to determine:

- Reliability of the critical components and subsystems.
- Overall reliability of the radio communication system.
- Availability of the critical components and subsystem.
- The sensitivity of the RAM parameters.
- The maintenance time and budget for corrective actions.
- The maintenance time and budget for preventive operation as a function of subsystem.

Note that only the tree first points have been worked in this paper.

The approach used in this work is based on the well-known standards STD-MIL which specify that reliability prediction utilizes a series model for system reliability evaluation for one or more components. It is permitted in case of critical components or according to the structure of a subsystem to use hot-standby and the redundancy items are modelled in parallel.

The reliability of the overall system is critically dependent upon the number of hours each component still functioning and given according of the Mean-Time Between Failures parameters (MTBF). The reliability of each component for a time duration t can be expressed as an exponential function depending on the relative MTBF. Tables 6.2 to 6.6 give the expected reliability and availability of each subsystem. Because the major parts of the system components are electronics, we note that both the availability and the reliability are similar due to negligible value of the MTTR (mean time to repair).
In the following tables, time is evaluated using one hour unit.

Table 6.2. Reliability of the microwave system

Time(h)	R(t)	(1 – R(t))
1000	0.8965	0.1035
2000	0.7987	0.2013
3000	0.713	0.287
4000	0.6422	0.3578
5000	0.5744	0.4256
6000	0.5136	0.4864
7000	0.4609	0.5391
8000	0.4187	0.5813
9000	0.3765	0.6235
10000	0.3343	0.6657

Table 6.3. Reliability of the radio constellation

Time(h)	R(t)	(1 – R(t))
2000	0.9339	0.0661
4000	0.8743	0.1257
6000	0.813	0.187
8000	0.7575	0.2425
10000	0.7087	0.2913
12000	0.66	0.34
14000	0.6136	0.3864
16000	0.5716	0.4284
18000	0.5322	0.4678
20000	0.4949	0.5021

Table 6.4. Reliability of the Ethernet System

Time(h)	R(t)	(1 – R(t))
13140	0.9575	0.0425
26280	0.9167	0.0833
39420	0.8811	0.1189
52560	0.8464	0.1536
65700	0.8114	0.1886
78840	0.777	0.223
91980	0.7471	0.2529
105120	0.7208	0.2792
118260	0.6928	0.3072
131400	0.8874	0.3326

Table 6.5. Reliability of the power system

Time(h)	R(t)	(1 – R(t))
13140	0.9999	0.0001
26280	0.9999	0.0001
39420	0.9998	0.0002
52560	0.9997	0.0003
65700	0.9994	0.0006
78840	0.9989	0.011
91980	0.9988	0.0012
105120	0.9986	0.0014
118260	0.9984	0.0016
131400	0.9984	0.0016

Table 6.6. Reliability of the VHF-THF system

Time(h)	R(t)	(1 – R(t))
13140	0.9999	0.0001
26280	0.9997	0.0003
39420	0.9994	0.0006
52560	0.9991	0.0009
65700	0.9989	0.0011
78840	0.9989	0.0011
91980	0.9988	0.0012
105120	0.9984	0.0016
118260	0.9982	0.0018

131400	0.9981	0.0019

Table 6.7. Reliability of the standard system

Time(h)	R(t)	(1 – R(t))
1000	0.8904	0.1096
2000	0.7944	0.2056
3000	0.7092	0.2908
4000	0.6353	0.3647
5000	0.565	0.435
6000	0.503	0.497
7000	0.4473	0.5527
8000	0.4001	0.5999
9000	0.3586	0.6414
10000	0.317	0.683

It is agreed that a site is modelled using a series of structures composed by the above five sub-systems.

The final table (Table 6.7) gives the overall reliability of the radio-communication network. Note that the detailed results cannot be included in the paper because the size of the final matrix is 156x156 nodes, while each node is composed by at least 20 components.

6.9 CONCLUSION

In this paper we have presented an efficient algorithm to evaluate network reliability whatever the complexity of the corresponding graph. This approach makes distinction between perfect and imperfect nodes and works according to the reality and exactitude of distributed complex systems and this avoids the doubt. The idea behind our development is to diminish the number of reduction and decomposition rules and thus the execution time. For that, we have shown that polygon-to-chain reductions are well-suited for computing network reliability. This feature gives a significant improvement in the execution time. We think that to be more accurate and to render the program faster, it is necessary to develop more efficient algorithms and tools to extract substructures like polygons and other topologies embedded in networks. This will be our next research. More, we have shown that radio communication systems can be modelling using Reliability block diagram and its reliability is evaluated using the algorithm which was introduced in this paper.

REFERENCES

Abraham, J. A., 1979, An improved algorithm for network reliability, *IEEE Transactions on Reliability*, Vol. 28, pp. 58-61.

Brace, K. S., Rudell, R. L. and Bryant, R. E., 1990, Efficient implementation of a BDD package, in *Proc. of the 27th ACM/IEEE Design Automation Conference*, pp. 40-45.

Bryant, R., 1986, Graph based algorithms for boolean function manipulation, *IEEE Transactions on Computers*, Vol. 35, NO. 8, pp. 677-691.

Choi, M. S. and Jun, C. H., 1995, Some variants of polygon-to-chain reductions in evaluating reliability of undirected network, *Microelectronics and Reliability*, Vol. 35, NO. 1, pp. 1-11.

Deo, N. and Medidi, M., 1992, Parallel algorithms for terminal-pair reliability, *IEEE Transactions on Reliability*, Vol. 41, NO. 2, pp. 201-209.

Dotson, W. P. and Gobien, J. O., 1979, A new analysis technique for probabilistic graphs, *IEEE Transactions Circuits and Systems*, Vol. Cas-26, NO. 10, pp. 855-865.

Fratta, L. and Montanari, U. G., 1973, A boolean algebra method for computing the terminal reliability in a communication network, *IEEE Transactions on Circuit Theory*, Vol. 20, NO. 3, pp. 203-211.

Hardy, G., Lucet, C. and Limnios, N., 2007, K-terminal network reliability measures with binary decision diagrams, *IEEE Transactions on Reliability*, Vol. 56, NO. 3, 506-515.

Heidtmann, K. D., 1989, Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability*, Vol. 38, NO. 3, pp. 305-311.

Institute of Electrical and Electronics Engineers, 1990, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, IEEE, New York.

Kuo, S. Y., Yeh, F. M. and Lin, H. Y., 2007, Efficient and exact reliability evaluation for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 56, NO. 2, pp. 288-300.

Lin, H. Y., Kuo, S. Y. and Yeh, F. M., 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, in *Proc. of the Eighth IEEE International Symposium on Computers and Communication*, Vo. 2, pp. 1341-1346.

Liu, H. H., Yang, W. T. and Liu, C. C., 1993, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronic Reliability*, Vol. 33, NO 4, pp. 599-613.

Locks, M. O. and Wilson, J. M., 1992, Note on disjoint products algorithms, *IEEE Transactions on Reliability*, Vol. 41, NO. 1, pp. 81-84.

Page, L. B. and Perry, J. E., 1988, A Practical Implementation of the factoring theorem for network reliability, *IEEE Transactions on Reliability*, Vol. 37, NO. 3, pp. 259-267.

Page, L. B. and Perry, J. E., 1989, Reliability of directed network using the factoring theorem, *IEEE Transactions on Reliability*, Vol. 38, NO. 5, pp. 556-562.

Resende, M. G. C., 1986, A program for reliability evaluation of undirected networks via polygon-to-chain reductions, *IEEE Transactions on Reliability*, Vol. 35, NO. 1, pp. 24-29.

Rudell, R., 1993, Dynamic variable ordering for ordered binary decision diagrams, in *Proc. of the 1993 IEEE/ACM inter. conference on Computer-aided design*, pp. 42-47.

Simard, C., 1996, Contribution Au Problème D'évaluation De La Fiabilité Des Réseaux, Mémoire de Maîtrise Sciences, *Laval University*, Québec, Canada.

Tarjan, R., 1972, Depth-first search and linear graph algorithms, *SIAM Journal of Computing*, Vol. 1, NO. 2, pp. 146-160.

Theolougou, O. R. and Carlier, J. G., 1991, Factoring and reduction for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 40, NO. 2, pp. 210-217.

Valian, L. G., 1979, The Complexity of enumerating and reliability problems, *SIAM Journal of Computing*, Vol. 8, NO. 3, pp. 410-421.

Wood, R. K., 1986, Factoring algorithms for computing k-terminal network reliability, *IEEE Transactions on Reliability*, Vol. 35, NO. 3, pp. 269-278.

Yan, L., Taha, H. A., Landers, T. L, 1994, A recursive approach for enumerating minimal cutsets in a network, *IEEE Transactions on Reliability*, Vol. 43, NO. 3, pp. 383-388.

Yoo, Y. B. and Deo, N., 1988, A comparison of algorithms for terminal-pair reliability, *IEEE Transactions on Reliability*, Vol. 37, pp. 210-215.

Chapter 7

Méthode de factorisation polygone-à chaine pour l'évaluation exacte de la fiabilité des réseaux dont les nœuds et les liens sont imparfaits

7.1 Résumé

Dans ce chapitre, nous proposons un modèle analytique pour l'évaluation exacte de la fiabilité des réseaux. L'algorithme suggère d'effectuer des remplacements de sous-structures topologiques identifiables dans le réseau par des sous-structures plus simples. Dans le cas général, ce processus s'appelle réduction polygone-à chaine. Le principe utilise les techniques de décomposition et de réduction basées sur le théorème de factorisation de Moore et Shannon. L'objectif est de montrer que même dans le cas ou les nœuds et les liens d'un réseau peuvent être sujets à des défaillances, l'évaluation de la fiabilité devient aisée une fois le réseau est prouvé décomposable. Pour cela, nous proposons des outils mathématiques et un algorithme qui permettent de calculer la fiabilité en temps linéaire.

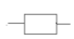
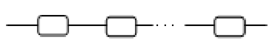
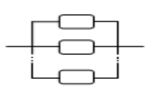
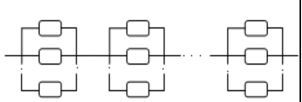
7.2 Introduction

Dans ce chapitre, nous nous intéressons au développement d'un modèle analytique pour l'évaluation de la fiabilité des réseaux. Nous utilisons le concept de réseau pour modéliser la structure de fonctionnement d'un système. Pour évaluer la fiabilité du système, il est nécessaire de connaitre la fiabilité de chacun de ses composants ainsi que sa structure. Dans la pratique, il n'existe pas une forme générale et unificatrice pour un tel calcul, sauf lorsque les structures sont des cas particuliers de type-- parallèle, série, série-parallèle, standby, k-parmi-n, etc. (voir table 7.1). Pour les réseaux, les méthodes principales d'évaluation de la fiabilité sont fondamentalement basées sur l'exploitation des chemins et des coupes minimales (Lin *et al.*, 2003), les méthodes de réduction, des sommes de produit-disjoints ((Abraham, 1979), (Heidtmann, 1989), (Locks, and Wilson, 1992), des diagrammes binaires de décision (Liu et al. 1993), (Bryant, 1986) et le concept d'inclusion-exclusion. Le problème d'évaluation de la fiabilité dans ces cas, est un problème NP-difficile (Valian, 1979) d'où la nécessité de développer de nouveaux outils de calcul pour à la fois évaluer la fiabilité du réseau et optimiser sa conception. Cependant, les méthodes proposées dans la littérature traitent de configurations particulières d'où leur performance a été évaluée pour de cas simples dont la valeur de la fiabilité est connue ou simple à évaluer.

En face de la complexité des réseaux modernes, il devient important de proposer de nouvelles techniques plus efficaces. Tout au long de ces dernières décennies, certaines méthodes ont été élaborées, comme celle de Moskowitz (1958), qui a introduit une solution qui utilise le théorème de décomposition par factorisation basée sur le théorème de Moore et Shannon (1956). Depuis lors, de nombreux articles ont été consacrés à ces idées comme ceux de Misra (1970), Murchland (1973), Rusenthal (1974), et Nakazawa (1976). D'autres travaux plus intéressants ont été publiés par Satyanarana and Chang (1983), Satyanarana and Wood (85), Wood (86), et dans lesquels leurs auteurs proposent un cadre unifié sur la base du théorème de factorisation permettant d'évaluer la fiabilité des réseaux dont les nœuds sont totalement fiables. Il est bien connu que dans la pratique, les réseaux deviennent très sensibles et imposent le fait de considérer non seulement les liens mais aussi les nœuds comme possiblement défaillants, c'est le cas par exemple des réseaux de télécommunications. Le but de ce chapitre est précisément d'élargir les travaux de Satyanarana *et* Wood. (1985) et ceux de Theologou *et* Carlier (1991), pour le cas de réseaux dont les nœuds et les liens peuvent être défaillants. Quelques chercheurs ont repris l'idée de réduction par factorisation polygone-à chaine (Choi., 1995), (Frattaand and Montanari, 1973), mais sans aller loin dans leur développement. Dans Theologou *et* Carlier (1991), les auteurs ont fait valoir que la décomposition polygone-à chaine ne peut être appliquée lorsque les nœuds d'un réseau sont défaillants. Nous allons à travers ce chapitre montrer comme l'a fait Wood (1986) pour les cas des réseaux avec des nœuds fiables, qu'il est possible d'obtenir un algorithme polynomial pour les cas ou les liens et les nœuds sont imparfaits. Ce dernier simplifie les graphes de type-1, de type-2 jusqu'au type-7 (voir de plus prés Satyanarana *et* Wood (1985), (Simard, 1996), (Rebaiaia *et al.,* 2009) pour comprendre cette notion de types) et ceci juste en appliquant le théorème de factorisation. Dans ce papier, il est question que les développements mathématiques prendrons en charge d'un coté les réductions polygone-à chaine de type 1 et de type 6 pour le cas ou seuls les liens sont défaillants et de type 1 et de type 7 lorsque les liens et les nœuds sont imparfaits. Cependant, il est utilise de savoir que pour le cas parfait Satyanarana *et* Wood (1985) ont présenté une démonstration qui traite uniquement la réduction polygone-à chaine du type 7 sans passer par le type 1. Nous nous sommes appliqués dans cette thèse à démontrer de deux façons différentes la validité de quelques théorèmes en rapport avec la factorisation et les travaux de Satyanarana and Wood que nous énoncerons par la suite. Pour le cas de la transformation polygone-à chaine des réseaux imparfaits, à notre connaissance, c'est le premier travail qui aborde de façon unique et claire cette problématique. Nous résumons dans la table 7.6 toutes les décompositions polygone-à chaine dans le cas des réseaux imparfaits.

Le contenu de ce chapitre est structuré comme suit : la section 2 présente les bases théoriques du principe de réduction, du théorème de factorisation et de la décomposition polygone-à chaine. A la section 3, les modèles traitant la factorisation polygone-chaine de type-1 et de type 6 sont détaillés, dans le cas ou seuls les liens sont sujets aux défaillances. La section 4, traite l'application du principe de la factorisation dans le cas ou les nœuds et les liens sont imparfaits. L'algorithme et un exemple pratique sont résumés dans la section 5 et une conclusion est présentée dans la section 6.

Table 7.1. Structures simples, fonctions de structures et fonctions de fiabilités

Structures	Fonction de structure	Forme mathématique
 A	$\Phi(X(t)) = x(t) = \begin{cases} 1 & si\ i\ fonctionne \\ 0 & si\ i\ est\ en\ panne \end{cases}$	$R(t)$
 A_1 A_2 A_n	$\Phi(X(t)) = \prod_{i=1}^{n} x_i(t)$	$R(t) = \prod_{i=1}^{n} R_i(t)$
	$\Phi(X(t)) = 1 - \prod_{i=1}^{n} (1 - x_i(t))$	$R(t) = 1 - \prod_{i=1}^{n} (1 - R_i(t))$
	$\Phi(X(t)) = 1 - \prod_{i=1}^{m} (1 - \prod_{j=1}^{n} x_{i,j}(t))$	$R(t) = 1 - \prod_{i=1}^{m} (1 - \prod_{j=1}^{n} R_{i,j}(t))$
	$\Phi(X(t)) = \prod_{i=1}^{m} (1 - \prod_{j=1}^{n} (1 - x_{i,j}(t)))$	$R(t) = \prod_{i=1}^{m} (1 - \prod_{j=1}^{n} (1 - R_{i,j}(t)))$
K parmi ***n***	$\Phi(X(t)) = \begin{cases} 1, si & \sum_{j}^{n} x_j(t) \geq k \\ 0, sinon & \end{cases}$	$R(t) = \sum_{j=k}^{n} C_n^j p^j (1-p)^{n-j}$

7.3 Préliminaires

7.3.1 Notions de base

Un réseau est un graphe $G = \ V,E$ où V est un ensemble fini de n nœuds (sommets) et E un ensemble fini de m arêtes qui sont les liens permettant de rendre possible la communication entre les nœuds. Si le graphe est orienté, on parlera d'arcs au lieu d'arêtes. On associe à chaque arc et à chaque nœud une probabilité de bon fonctionnement et on considère que ces dernières sont statistiquement indépendantes. Nous adoptons la notation ***s*** et ***t*** pour définir le choix du sommet source et du sommet de destination et l'on parlera alors d'un problème 2-terminaux, dit aussi terminal-paire (Deo and Medidi, 1992), (Yoo and Deo, 1988), (Dotson, and Gobien, 1979). Pour certains réseaux, on peut s'intéresser à l'évaluation de la probabilité que K- nœuds soient reliés entre eux, on parlera alors de la fiabilité K-terminaux notée $R(G_k)$ qui représente la probabilité que tous les K-nœuds soient reliés entre eux par une arborescence (noter que les K-nœuds sont dessinés par des rond pleins). La généralisation d'une telle notion parait nécessaire pour certaines classes de problèmes, dits tous-terminaux. Notez qu'un 2-terminaux suffit amplement à définir la fiabilité du réseau.

Dans la pratique deux situations peuvent se produire lorsqu'il est question de calculer la fiabilité d'un réseau. Soit que ce réseau est réductible et qu'on lui applique successivement une série de règle de réductions simples du type série, série-parallèle, réduction de degré-2, delta-étoile et vice-versa, ou bien, dans le cas contraire le graphe est alors irréductible (voir figure 7.1), on procède alors par l'application d'une série de réductions dites polygones-chaines (Theolougou and Carlier, 1991), (Wood, 1986).

Figure 7.1. Gauche : Graphe réductible Droite : Graphe irréductible

Pour mieux expliquer la problématique de la factorisation, nous présentons dans ce qui suit, certaines définitions et formalismes qui seront utilisés comme éléments de base à notre contribution. Toute fois, il est important de noter que les opérations de réductions série-parallèle sur un graphe réductible peuvent être calculées en un temps linéaire et ceci dépendamment de la taille du graphe (Wood. 1985).

7.3.1.1 Principe du Théorème de Factorisation

Les méthodes de factorisation dites aussi de réduction, utilisent le théorème de factorisation de Moore & Shannon. Elles consistent essentiellement à décomposer un graphe en faisant des hypothèses sur l'état d'un composant, jusqu'à ce que l'on obtienne des graphes à configurations simples. Le théorème des probabilités totales permet alors de calculer la fiabilité du graphe à partir de sous-graphes obtenus. L'idée derrière ce processus étant de postuler dans un graphe qu'un arc ***e*** est bon $(\Phi(X(t)) = 1|x_e = 1)$ revient à dire que la communication à travers cet arc est assurée. Le cas contraire s'écrit $(\Phi(X(t)) = 1|x_e = 0)$. Dans un graphe orienté, une communication parfaite entre deux nœuds est équivalente à les fusionner en un seul nœud, et l'impossibilité de communiquer à travers un arc entraine sa suppression.

Considérons un graphe *G* dans lequel on choisi un composant *j* aléatoirement. Le théorème des probabilités totales permet pour ce composant *j*, d'exprimer la fiabilité du graphe *G* par :

$$R = \Pr(\Phi(X(t)) = 1) = \Pr(\Phi(X(t)) = 1|x_j = 1) \times \Pr(x_j = 1) + \Pr(\Phi(X(t)) = 1|x_j = 0) \times \Pr(x_j = 0)$$

Si l'on substitue la probabilité $\Pr(x_j = 1)$ par R_j et $\Pr(x_j = 0)$ par $(1 - R_j)$ qui sont respectivement les probabilités de bon fonctionnement et de défaillance, Il s'ensuit, que :

$$R = \Pr(\Phi(X(t)) = 1) = \Pr(\Phi(X(t)) = 1|x_j = 1) \times R_j + \Pr(\Phi(X(t)) = 1|x_j = 0) \times (1 - R_j) \quad (7.0)$$

Ce processus de décomposition se poursuit autant de fois qu'il est nécessaire, autrement dit, jusqu'à ce qu'une structure simple soit trouvée (si elle existe) et dont la fiabilité est facile à évaluer. Il est à noter que le choix de certains composants peut parfois faciliter le nombre de décompositions, ce qui nécessite d'utiliser une heuristique pour le choix. Cette façon de raisonner permet de postuler qu'un lien entre deux nœuds est équivalent à fusionner ces deux nœuds en un seul. Le contraire, conduit à supprimer ce lien.

Notons que l'équation (7.0) peut aussi s'écrire sous la forme de l'équation (7.1) en supposant que si $e_i = (u,v)$ est un lien du graphe G_K et, F_i et $\overline{F}_i$ dénotent respectivement que l'événement e_i (le composant) soit qu'il fonctionne, soit qu'il est défaillant.

$$R(G_K) = p_i R(G_K | F_i) + q_i R(G_K | \overline{F}_i) = p_i R(G'_{K'}) + q_i R(G'_{K'}) \tag{7.1}$$

A partir du théorème des probabilités totales, certains auteurs ont énoncé simplement le théorème de factorisation, qui permet d'écrire que pour un composant ***e***, choisi arbitrairement, la fiabilité du réseau G est exprimée par :

$$R(G) = p_e . R(G * e) + (1 - p_e) . R(G - e) \tag{7.2}$$

où

$R\ G$: la fiabilité du réseau G.
$R\ G * e$: la probabilité que le système fonctionne lorsque le composant ***e*** fonctionne (l'arc ***e*** est contracté).
$R\ G - e$: la probabilité que le système fonctionne lorsque le composant ***e*** est défaillant (l'arc ***e*** est éliminé).

Et tel que :

$$G * e_i = (V - u - v + w, E - e_i), \qquad w = u \cup v,$$

$$G - e_i = (V, E - e_i),$$

$$K' = \begin{cases} K & if \quad u, v \notin K \\ K - u - v + w & if \quad u \in K \quad or \quad v \in K \end{cases}$$

$$K'' = K$$

Pour illustrer le théorème de factorisation nous présentons un graphe simple dit *Bridge network* ou *Pont de Königsberg* (figure 7.2). Pour cela, supposons que le composant 3 étant l'élément pivot. Les différentes étapes de décomposition sont illustrées dans la figure suivante.

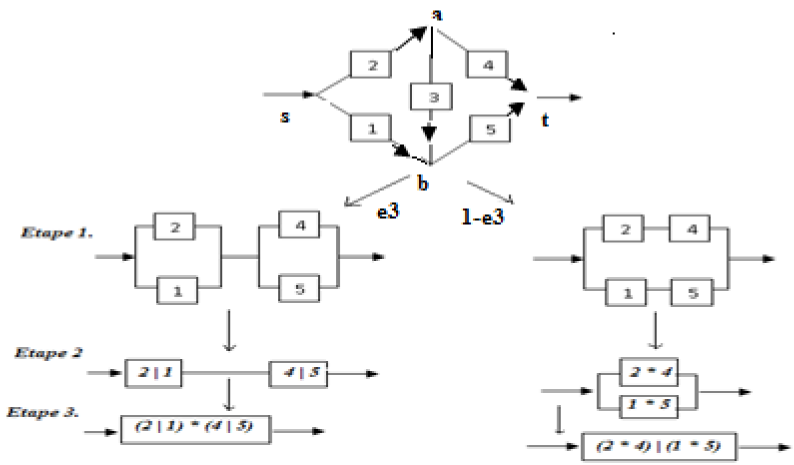


Figure 7.2. Décomposition du réseau *Pont de Königsberg*.

Connaissant la fiabilité $p_1, p_2, p_3, p_4\ et\ p_5$ de chaque composant ainsi que le composant pivot (ici le composant 3), la fiabilité du système peut être facilement évaluée à partir de relations résultantes identifiées par les feuilles de l'arbre de décomposition.

$$R(G) = p_3 R(G{*}e_3) + (1 - p_3) R(G - e_3) = p_3 [1 - (1 - p_1)(1 - p_2)][1 - (1 - p_4)(1 - p_5)] + (1 - p_3)[1 - (1 - p_2 p_4)(1 - p_1 p_5)]$$

Remarque 1: Pour identifier l'expression équivalente, nous avons procédé par séparation du graphe en deux parties, puis de proche en proche nous avons effectué des compositions parallèles et séries (voir figure 7.2). Le but est de ramener le réseau vers une structure simple à identifier. Le processus s'arrêtera lorsqu'il ne sera plus possible de faire des réductions simples.

Remarque 2 : Compte tenu de la complexité de cette méthode qui est de nature exponentielle, il est souhaitable de réduire au maximum la taille du graphe avant d'appliquer le théorème de factorisation. Pour cela les réductions suivantes sont nécessaires.

Le principe du théorème de factorisation permet de calculer la fiabilité de n'importe quel type de réseau, à condition qu'il soit décomposable. Donc, à chaque application du théorème de factorisation, on décompose le graphe en deux parties de tailles légèrement réduites, à savoir, $G * e$ qui est le graphe G avec un nœud et une liaison en moins, et $G - e$

avec une liaison en moins (arc, arête). Si n est le nombre de liaisons dans un réseau G, alors dans le pire des cas, on utilisera le théorème de factorisation $(2^n - 1)$ fois.

Nous constatons que l'application du théorème de factorisation engendre une complexité exponentielle, dans ce cas de figure, il est important pour l'accélération des calculs de procéder à des réductions sur la taille du graphe.

7.3.2 Réductions et factorisation

Le principe de la réduction d'un graphe de fiabilité avant l'application du théorème de factorisation est de diminuer autant que possible le nombre des nœuds et des arêtes à condition que la valeur de la fiabilité reste invariable. Cela, conduit à générer un nouveau graphe G' équivalent en termes de fiabilité au graphe initial G. Donc, il s'agit de remplacer une structure topologique complexe par une autre plus simple qui préserve les mêmes propriétés de fiabilité. C'est-à-dire que si Ω est une constante de transformation propre à chaque formule de transformation. Dans le cas général nous définissons alors la relation suivante :

$$R\ G\ = \Omega\, R\ G' \tag{7.3}$$

Il est important de noter que le processus de réduction est d'une complexité polynomiale contrairement à la décomposition qui est exponentielle. Ce processus doit procéder dans un premier temps, par une succession de réductions simples appliquées à des graphes dits série-parallèles. Elles se définissent comme suit :

7.3.2.1 Réductions séries-parallèles

Le processus de réduction tient compte précisément des données relatives aux fiabilités et celle des K-nœuds du graphe, alors le remplacement parallèle et le remplacement série ne font que modifier la structure topologique du graphe d'une forme complexe vers une forme simple à partir de laquelle, la fiabilité du graphe est déduite avec moins d'effort de calcul.

Les réductions simples les plus fréquentes se résument comme suit :

Réduction série. Soient $e_a = (u, v)$ et $e_b = (v, w)$ deux arêtes en série définies dans G_K et, tel que le $degré\ \ v\ \ = 2$ et $v \notin K$. Une réduction série obtient G' par le remplacement de e_a et e_b par une seule arête $e_c = (u, w)$ de fiabilité $p_c = p_a p_b$, et qui définit Ω = 1 et *K'* = *K*, et où $R\ \ G\ \ = \Omega R(G')$.

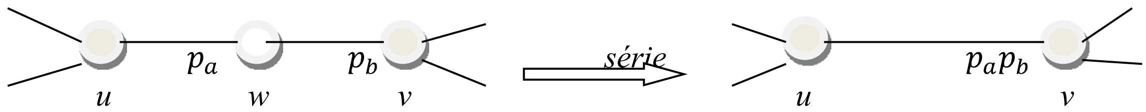


Figure 7.3. Réduction série

Réduction parallèle. Soient $e_a = (u, v)$ et $e_b = (u, w)$ deux arêtes parallèles dans G_K et supposons que $p_i = 1 - q_i$ (i = a or b), la fiabilité associée à l'arête i. Une réduction parallèle permet de remplacer e_a et e_b par une seule et unique arête $e_c = (u, v)$ de fiabilité $p_c = 1 - q_a q_b$, et qui définit Ω = 1 et *K'* = *K* et où $R\ \ G\ \ = \Omega R(G')$.

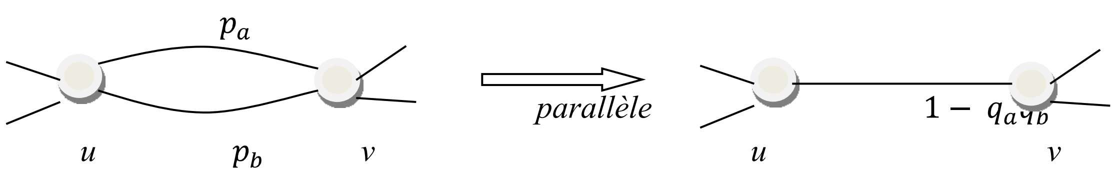


Figure 7.4. Réduction parallèle

Réduction de Degré-deux. Soient $e_a = (u, v)$ et $e_b = (v, w)$ deux arêtes en série dans G_K, telles que $v \neq w$, $degré\ \ v\ \ = 2$ et $u, v\ et\ w \in K$. Une réduction de degré-deux crée G' par le remplacement de e_a et e_b par une seule et unique arête $e_c = (u, v)$ de fiabilité $p_c = \frac{p_a p_b}{1 - q_a q_b}$, avec $K' = K - v$, et $\Omega = 1 - q_a q_b$.

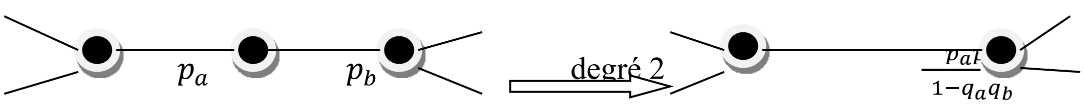


Figure 7.5. Réduction de degré deux

Réduction delta-à étoile. La réduction delta-à étoile consiste à remplacer une structure topologique delta par une structure étoile. Les sommets de la structure delta doivent être tous de même type : tous des K-nœuds ou tous des non-K-nœuds. Le sommet ainsi rajouté pour le besoin de la transformation vers la structure étoile n'appartient pas à K et sa probabilité est calculée comme suit (figure 7.6):

$$p_x = \frac{\alpha}{\alpha + \beta_1}, \quad p_y = \frac{\alpha}{\alpha + \beta_2}, \qquad p_z = \frac{\alpha}{\alpha + \beta_3}$$

$$p_{u_0} = \frac{(\alpha + \beta_1)(\alpha + \beta_2)(\alpha + \beta_3)}{\alpha^2}$$

Avec $\alpha = p_a p_b + p_a p_c + p_b p_c - 2 p_a p_b p_c$

$\beta_1 = q_a q_b p_c$, $\beta_2 = q_a p_b p_c$, $\beta_3 = p_a q_b q_c$

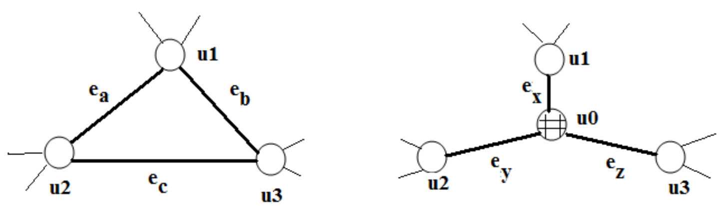


Figure 7.6. Réduction delta-à étoile

Réductions polygone-à chaine. Dans un graphe G, une chaine est une séquence alternée entre les nœuds et les arêtes du graphe. Sa longueur est le nombre d'arêtes qui la compose et elle est au moins égale à 1. Aussi, les sommets internes sont tous de degré 2, et les sommets aux extrêmes sont de degré supérieur à 2. Si le graphe admet au moins deux chaines χ_1 *et* χ_2 de longueur respectivement l_1 et l_2, et si ces deux chaines ont en commun deux nœuds en terminaux u et v, alors $\Delta = \chi_1 \cup \chi_2$ est un polygone de longueur $l_1 + l_2$. Autrement dit, le polygone forme un cycle dont deux sommets sont de degré supérieur à 2. Il est aussi vrai que deux chaines parallèles délimitées entre deux nœuds peuvent être substituées et donc remplacées par une nouvelle chaine. Cette opération est dite réduction *polygone-à chaine*. Ce qui suggère qu'une transformation polygone-à chaine, consiste à remplacer un polygone par une chaine. Aussi, comme les polygones se diversifient par la forme de leur structure, c'est-à-dire le nombre d'arêtes les composant, Satyanarayana *et*

Wood (1985), ont énuméré 7 types de polygones pouvant être remplacés par une chaine. C'est ainsi, que par une série de transformations polygone-à chaine qui peuvent être entrecoupées par des réductions série, parallèle, delta-étoile et étoile-delta que tout réseau qui admet des sous structures de type polygone, dans le meilleur des cas, peut être transformé en une structure très simple (par exemple une chaine) et par la même, sa fiabilité est réduite par application des formules ainsi obtenues.

Notons que la table 7.1 (page 823) publiée par Satyanarayana *et* Wood (1985) et reprise dans Wood (1986) présente les réductions polygone-à chaine pour les réseaux parfaits. Néanmoins, les auteurs ont présenté les rudiments du calcul basé sur le théorème 1, donné à la page 823 et sa démonstration. Ce théorème fournit un cadre général sur lequel nous pouvons déduire les transformations polygones-chaine des 7 figures. Il est utile de constater que le processus de transformation n'est pas aussi aisé que les auteurs pensent sans la reproduction de toutes les étapes de transformation. Ce travail va être fait par nous dans le cadre de cet article. Il constitue un apport très important pour montrer que sans passer par les détails de l'application du théorème de factorisation, l'on serait incapable de trouver toutes les formules de transformation énoncées dans la table 1 de Satyanarayana *et* Wood (1985). Notons aussi que les travaux de Satyanarayana *et* Wood prennent en charge uniquement les réseaux non-orientés dont seules les arêtes sont susceptibles de représenter les défaillances et le bon fonctionnement. La réalité est toute autre, car les systèmes actuels sont plus complexes et donc les nœuds peuvent aussi subir des défaillances. Plusieurs travaux ont été réalisés dans ce sens et des algorithmes énoncés. Parmi ceux-là, Théologou *et* Carlier dans (1991) ont présenté un algorithme qui prend en charge cette problématique. Nous y reviendrons un peu plus tard sur ce cas en proposant un schéma de démonstration qui ne se trouve dans aucun travail (à notre connaissance) fait en dehors des travaux de Simard (1996) et Rebaiaia *et al.* (2009),

Dans ce qui suit, nous présentons une dynamique de transformation qui détaille contrairement à Satyanarayana *et* Wood toutes les opérations de réduction. Pour cela, nous commencerons par le cas polygone-à chaine de type 1 et nous énoncerons le théorème 1. Nous ferrons aussi le parallèle en considérant le cas de polygone-à chaine lorsque les nœuds et les arêtes peuvent être sujets à des défaillances. On dira alors que les arêtes et les nœuds sont imparfaits. Pour cela nous énoncerons le théorème 7.2 pour le cas de la transformation de polygone-à chaine de type 6.

7.4 Factorisation polygone-à chaine- Cas des liens imparfaits et nœuds parfaits

7.4.1 Réduction polygone-à chaine de type 1

Pour démontrer les résultats d'une telle transformation, considérons le cas d'une réduction polygone-à chaine de type 1 (il est conseillé de voir Satyanarayana *et* Wood (1985) ou Rebaiaia *et al*., (2009). Dans ce cas, le graphe G' qui est l'équivalent du graphe G, est obtenu en remplaçant le polygone de type 1 (triangle) par la chaine qui possède deux arcs e_1' *et* e_2' avec les fiabilités respectives p_1' *et* p_2' (voir figure 7.7). La relation générale qui relie la fiabilité du graphe G et G' est donnée par $R(G) = \Omega R(G')$. Le problème dans ce cas est de déterminer le triplet (Ω, p_1', p_2').

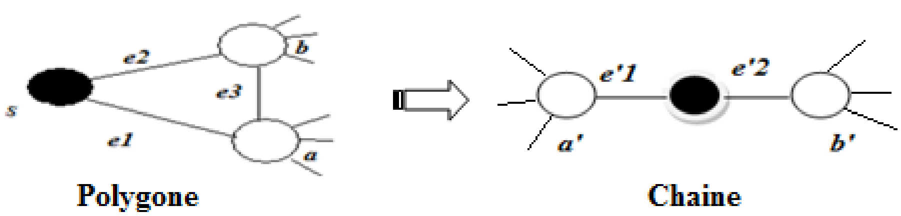


Figure 7.7. Réduction polygone-chaine de type 1

Pour le faire, nous énonçons le théorème suivant :

Théorème 7.1.

Supposons qu'un graphe G contienne un polygone de type-1 tel que présenté dans la figure 7.7 (Gauche). Soit G' le graphe obtenu par transformation du graphe G par remplacement des arêtes e_1 *et* e_2 de probabilités p_1 *et* p_2 par les arêtes e_1' *et* e_2' de fiabilité p_1' *et* p_2' et soit Ω un facteur de multiplication. Alors

$R(G) = \Omega R(G')$

où $\Omega = (\delta + A)(\delta + B)/\delta$, $A = q_1 p_2 q_3$, $B = p_1 q_2 q_3$, $\delta = p_1 p_2 p_3 [1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3}]$,

$p'_1 = \delta/(\delta + A)$ et $p'_2 = \delta/(\delta + B)$.

Preuve :

La première étape consiste à appliquer successivement la procédure de factorisation sur les arcs pivots e_1, e_2 et e_3. Nous procédons comme suit :

Pivotons tout d'abord sur l'arc e_1 (voir la figure 7.8) tout en appliquant l'équation (1), la fiabilité du graphe devient :

$$R\ G\ = p_1 R(G * e_1) + q_1\ R\ G - e_1 \tag{7.4}$$

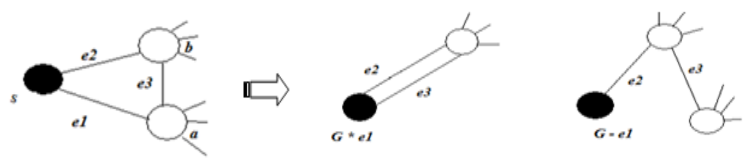


Figure 7.8. Première étape du processus de factorisation

Puis, décomposons sur l'arc e_2, nous obtenons la relation (7.5):

$$\begin{aligned} R(G) = {} & p_1\big[p_2 R\big((G * e_1) * e_2\big) + q_2 R\big((G * e_1) - e_2\big)\big] \\ & + q_1\big[p_2 R\big((G - e_1) * e_2\big) + q_2 R\big((G - e_1) - e_2\big)\big] \end{aligned} \tag{7.5}$$

Nous pouvons remarquer que la fiabilité du graphe $(G - e_1\ - e_2)$ est nulle. En effet, en retirant les arcs $e_1\ et\ e_2$ de G, le K-nœud se trouve complètement isolé, par conséquent cette opération génère une fiabilité nulle. La relation (7.5) précédente devient :

$$R\ G\ = p_1 p_2 R\ \ G * e_1\ * e_2) + p_1 q_2 R\ (G * e_1\ - e_2\ + p_2 q_1 R\ (G - e_1\ * e_2) \tag{7.6}$$

Appliquons de nouveau le théorème de factorisation aux graphes réduits $G * e_1\ - e_2$ et $G - e_1\ * e_2$ en considérant l'arc e_3. La relation (7.6) est en transformée en (7.7) :

$$\begin{aligned} R(G) = {} & p_1 p_2 R\big((G * e_1) * e_2\big) + p_1 q_2 [p_3 R\big((G * e_1) - e_2\big) * e_3) + q_3 R(((G * e_1) - e_2) - \\ & e_3)] + p_2 q_1 [p_3 R((G - e_1) * e_2) * e_3) + q_3 [R((G - e_1) * e_2) - e_3)] \\ = {} & p_1 p_2 R\big((G * e_1) * e_2\big) + p_1 q_2 p_3 R\big((G * e_1) - e_2\big) * e_3) + q_3 p_1 (1 - p_2) R(((G * e_1) \\ & - e_2) - e_3)] + p_2 q_1 p_3 R(((G - e_1) * e_2) * e_3) + p_2 q_1 (1 - p_3) R(((G - e_1) * e_2) \\ & - e_3) \end{aligned} \qquad ((7.7)$$

Notons que nous n'avons pas pu éliminer l'arc e_3 par décomposition du graphe $G * e_1 * e_2$, puisque cet arc ne fait plus partie du graphe correspondant à la contraction des arcs $e_1$et e_2.

L'équation précédente (7.7) est l'expression de la fiabilité du graphe G en fonction de graphes réduits n'ayant plus les arcs du polygone. Comme les graphes correspondant à, $(G * e_1) * e_2$, $\left((G - e_1) * e_2\right) * e_3$ et $(G * e_1) - e_2) * e_3$ sont identiques, il en résulte que les fiabilité de ces graphes sont aussi identiques. Donc l'expression (7.7) devient :

$$R(G) = [p_1p_2 + p_1q_2p_3 + q_1p_2p_3]R\left((G * e_1) * e_2\right) + p_1q_2q_3R(((G * e_1) - e_2) - e_3) + q_1p_2q_3R(((G - e_1) * e_2) - e_3) \tag{7.8}$$

Par une démarche identique, on peut exprimer la fiabilité du graphe G' en pivotant successivement sur les arcs e_1' et e_2' et donc la fiabilité du graphe G' s'écrit :

$$R(G') = p_1'\left[p_2'R\left((G' * e_1') * e_2'\right) + (1 - p_2')R((G' * e_1') - e_2')\right] + (1 - p_1')\left[p_2R\left((G - e_1') * e_2')\right)\right] \tag{9.9}$$

Il est nécessaire dans le cas actuel d'exprimer la relation permettant de relier la fiabilité de G' à celle de G. Nous pouvons remarquer qu'il existe une similitude entre les formules et il en découle que:

$$R\left((G * e_1) * e_2\right) = R\left((G' * e_1') * e_2')\right)$$

$$R\left((G * e_1) - e_2\right) - e_3) = R\left((G' * e_1') - e_2')\right)$$

$$R\left((G - e_1) * e_2\right) - e_3) = R\left((G' - e_1') * e_2')\right)$$

Comme la formule générale de réduction est donnée par $R\ G\ = \Omega R(G')$, on obtient un système de trois équations à trois inconnues (Ω, p_1', p_2') et qui est :

$$\begin{cases} p_1p_2 + p_1q_2p_3 + q_1p_2p_3 = \Omega p_1'p_2' \\ q_1p_2q_3 = \Omega q_1'p_2' \\ p_1q_2q_3 = \Omega p_1'q_2' \end{cases} \tag{7.10}$$

En résolvant ce système on obtient une formule de réduction qui permet de remplacer un polygone de type 1 par une chaine de longueur deux. Les fiabilités relatives aux arcs de la chaine sont :

$$\begin{cases} p_1' = \delta/(\delta + A) \\ p_2' = \delta/(\delta + B) \\ \Omega = (\delta + A)(\delta + B)/\delta \end{cases} \tag{7.11}$$

De même

$$\begin{cases} A = q_1 p_2 q_3 \\ B = p_1 q_2 q_3 \\ \delta = p_1 p_2 p_3 [1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3}] \end{cases} \tag{7.13}$$

Finalement, nous notons qu'il est possible de dériver une topologie simplifiée à partir d'une autre plus complexe tout en préservant l'expression de la fiabilité en tant que telle, et ceci après l'application d'une série de réductions. Le problème qui risque de surgir étant, l'automatisation de la reconnaissance d'une certaine topologie dont les calculs se déduisent très facilement. L'idée est très bénéfique à condition de concevoir des algorithmes puissants qui permettent de sauter facilement une telle phase critique du processus de calcul, ou du moins de déterminer les moyens nécessaires de chercher les équivalences entre structures.

7.5 Réduction polygone-à chaine de type 6

Nous pouvons à présent énoncer le théorème de Satyanarayana *et* Wood (1985) qui généralise le calcul uniquement pour le cas ou seules les arêtes sont imparfaites.

Théorème 7.2.

Supposons que G_K contienne un polygône de type J (1≤J≤7). Soit, $G'_{K'}$, le graphe obtenu après le remplacement du polygône Δj de G_K de chaine χ_j, et soit Ωj le facteur de multiplication. Alors, $R\ G_K = \Omega_j R(G'_{K'})$.

La démonstration du théorème se trouve dans Satyanarayana *et* Wood (1985) aux pages 824-825.

A partir du théorème 7.2 et en cheminant de la même manière que pour le cas de réduction-à chaine de type 1 lorsque seuls les liens sont sujets à des défaillances, nous énonçons le théorème suivant :

Théorème 7.3.

Supposons qu'un graphe G contienne un polygone de type-6 tel que présenté dans la figure 8 (a). Soit G' le graphe obtenu par transformation du graphe G par remplacement des arêtes e_1, e_2, e_3, e_4 *et* e_5 de probabilités p_1, p_2, p_3, p_4 *et* p_5 par les arêtes, e_r, e_s *et* e_t de fiabilité p_r, p_s *et* p_t et soit Ω un facteur de multiplication. Alors

$$R\ G\ = \Omega R(G') \text{ et } p_r = \frac{D}{\alpha + D}\ ;\ p_s = \frac{D}{\beta + D}\ ;\ p_t = \frac{D}{\delta + D}\ ;\ \Omega = \frac{(A+D)(B+D)(C+D)}{D^2}$$

$$A = \Omega . q_r p_s p_t\ ;\ C = \Omega . p_r p_s q_t\ ;\ B = \Omega . p_r q_s p_t\ ;\ D = \Omega . p_r p_s p_t$$

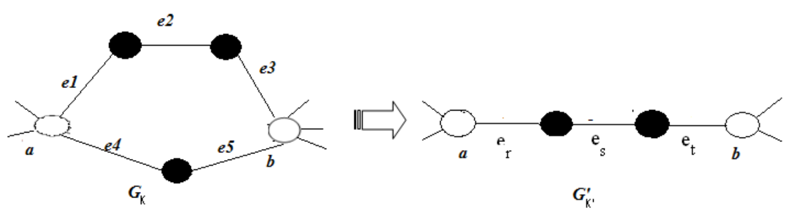


Figure 7.9. Factorisation polygone-à chaine de type-6

Preuve :

Adoptons une démarche différente que pour le cas polygone-à chaine de type 1. Au lieu de rappeler tous les graphes induits suite aux décompositions ainsi que l'expression de leurs fiabilités, nous résumerons le processus de factorisation en fournissant les graphes non-défaillants (les autres provoquant la nullité de la fiabilité sont écartés), les événements et la valeur des fiabilités relatives. La table 7.2, résume les résultats de la décomposition polygone-à chaine de type 6. Il est simple de constater que le nombre de combinaisons des événements qui forment l'état du système est égal à 2^5. Cependant, nous constatons que chaque état non-défaillant induira un nouveau graphe constitué du nombre correspondant de K-nœuds. Il y en aura dans ce cas 4 graphes comme dans la figure 7.10 avec les états et les probabilités dans chaque cas.

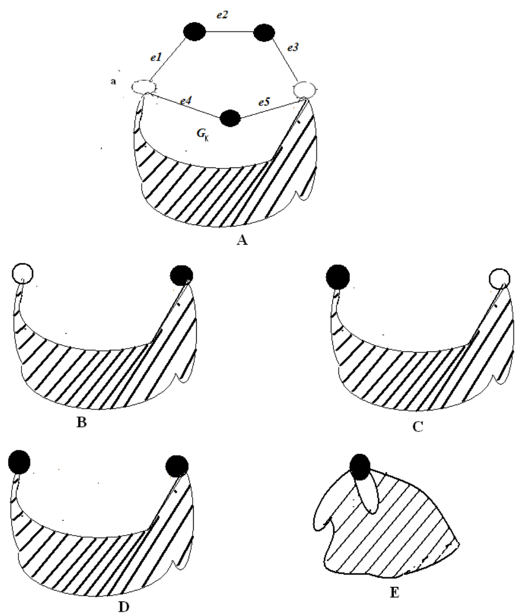


Figure 7.10. Graphe G renfermant un polygone de type-6 (A) et ces 4 sous-graphes induits non-défaillants (B, C, D et E)

A partir des graphes de la figure 7.10, nous déterminons les états et les probabilités correspondant à chaque graphe induit. Nous supposons toujours que F_i et $\overline{F}_i$ dénotent respectivement que l'événement e_i (le composant) soit qu'il fonctionne, soit qu'il est défaillant et chaque p_i (resp. $q_i = 1 - p_i$) étant la probabilité de non-défaillance (resp. défaillance). Toutes les formules sont reportées dans la table suivante :

Table 7.2. États de non-défaillants et les probabilités des graphes ainsi déduits.

Graphe	États	Probabilités
B	$\overline{F_1}F_2F_3\overline{F_4}F_5$	$\alpha = q_1p_2p_3q_4p_5$
C	$F_1F_2\overline{F_3}F_4\overline{F_5}$	$\delta = p_1p_2q_3p_4q_5$
D	$F_1\overline{F_2}F_3\overline{F_4}F_5$; $F_1\overline{F_2}F_3F_4\overline{F_5}$; $F_1F_2\overline{F_3}\,\overline{F_4}F_5$; $\overline{F_1}F_2F_3F_4\overline{F_5}$	$\beta = p_1q_2p_3q_4p_5 + p_1q_2p_3p_4q_5 + p_1p_2q_3q_4p_5 + q_1p_2p_3p_4q_5$ $= p_1q_2p_3(q_4p_5 + p_4q_5) + p_2(q_1p_3p_4q_5 + p_1q_3q_4p_5)$
E	$F_1F_2F_3F_4F_5$; $\overline{F_1}F_2F_3F_4F_5$; $F_1\overline{F_2}F_3F_4F_5$; $F_1F_2\overline{F_3}F_4F_5$; $F_1F_2F_3\overline{F_4}F_5$; $F_1F_2F_3F_4\overline{F_5}$	$\gamma = p_1p_2p_3p_4p_5 + q_1p_2p_3p_4p_5 + p_1q_2p_3p_4p_5 +$ $p_1p_2q_3p_4p_5 + p_1p_2p_3q_4p_5 + p_1p_2p_3p_4q_5$ $= p_1p_2p_3p_4p_5(1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4} + \frac{q_5}{p_5})$

Finalement, en regroupant les expressions similaires et en supprimant d'autres qui induisent la nullité, l'expression de fiabilité du polygone de type 6 ainsi réduite est comme suit :

$$R(G_K) = \alpha.R(G_{B,K_B}) + \beta.R(G_{C,K_C}) + \delta.R(G_{D,K_D}) + \gamma.R(G_{E,K_E})$$

Le pivotage successif sur les liens e_1, e_2, e_3, e_4 et e_5, transforme le graphe G_K à une chaine $G'_{K'}$ (voir figure 7.9). Cependant lorsqu'on utilise le théorème de factorisation sur $G'_{K'}$ pour déterminer les formules d'équivalences des termes, nous identifions sur $G'_{K'}$ quatre graphes non-défaillants, les états ainsi que les probabilités correspondants. En supposant toujours que F_r, F_s and F_t sont les événements relatifs aux liens r, s et t (figure 7.9. (b)), les résultats sont résumés dans la table 7.3.

Table 7.3. États non-défaillants et les probabilités des graphes induits $G'_{K'}$.

Graphe	**States**	**Probabilités**
B'	$\overline{F_r} F_s F_t$	$\alpha' = q_r p_s p_t$
C'	$F_r F_s \overline{F_t}$	$\beta' = p_r p_s q_t$
D'	$F_r \overline{F_s} F_t$	$\delta' = p_r q_s p_t$
E'	$F_r F_s F_t$	$\gamma' = p_r p_s p_t$

En regroupant les termes équivalents, la fiabilité du graphe ainsi réduit est :

$$\begin{aligned} R(G'_{K'}) &= \alpha'.R(G'_{B',K'_{B'}}) + \beta'.R(G'_{C',K'_{C'}}) + \delta'.R(G'_{D',K'_{D'}}) + \gamma'.R(G'_{E',K'_{E'}}) \\ &= q_r p_s p_t.R(G'_{B',K'_{B'}}) + p_r p_s q_t.R(G'_{C',K'_{C'}}) + p_r q_s p_t.R(G'_{D',K'_{D'}}) + p_r p_s p_t.R(G'_{E',K'_{E'}}) \end{aligned}$$

Utilisons présentement la relation $R(G_K) = \Omega R(G'_{K'})$, nous pouvons alors identifier les coefficients α, β, δ et γ comme suit :

$$\begin{aligned} &\alpha.R(G_{B,K_B}) + \beta..R(G_{C,K_C}) + \delta.R(G_{D,K_D}) + \gamma.R(G_{E,K_E}) = \Omega.[q_r p_s p_t.R(G'_{B',K'_{B'}}) \\ &+ p_r p_s q_t.R(G'_{C',K'_{C'}}) + p_r q_s p_t.R(G'_{D',K'_{D'}}) + p_r p_s p_t.R(G'_{E',K'_{E'}})] \end{aligned}$$

Égalisons les termes des deux cotés de l'équation $R(G_K) = \Omega R(G'_{K'})$, il s'ensuit l'expression des égalités suivantes :

$$\alpha = \Omega.q_r p_s p_t \;;\quad \beta = \Omega.p_r q_s p_t \;;\quad \delta = \Omega.p_r p_s q_t \;;\quad \gamma = \Omega.p_r p_s p_t$$

Finalement après résolution du système d'équations nous obtenons :

$$p_r = \frac{\gamma}{\alpha + \gamma} \;;\quad p_s = \frac{\gamma}{\beta + \gamma} \;;\quad p_t = \frac{\gamma}{\delta + \gamma} \;;\quad \Omega = \frac{(\alpha + \gamma)(\beta + \gamma)(\delta + \gamma)}{\gamma^2} \;,$$

et tel que les expressions réelles de α, β, δ et γ sont tirées de la table 2 (troisième colonne).

7.6 Factorisation polygone-à chaine- Cas des liens et nœuds imparfaits

7.6.1 Principes de la décomposition en présence de nœuds imparfaits

Les hypothèses suivantes ont été retenues :

- Les arêtes et les nœuds sont sujets à des défaillances
- Les défaillances sont s-indépendantes avec des probabilités connues
- Tous les K-nœuds du graphe sont parfaitement fiables.

Notons que dans le cas des réseaux imparfaits, certaines modifications doivent être apportées au sens même de la factorisation compte-tenu du principe d'indépendance qui doit impérativement être respecté. Cependant les réductions suivantes doivent prévaloir :
Réduction série. Soient $e_a = (u, v)$ et $e_b = (v, w)$ deux arêtes en série définies dans G_K et, tel que le $degré\ v\ = 2$ et $v \notin K$. Une réduction série obtient G' par un remplacement de e_a et e_b par une seule arête $e_c = (u, w)$ de fiabilité $p_c = p_a p_b p_v$.

Réduction parallèle. La réduction parallèle reste inchangée.

Réduction de Degré-un. Quand un sommet imparfait *v* est adjoint à un K-nœud de *degré 1* suivi du lien e_a on a : $\Omega = p_a p_v$.

Réduction de Degré-deux. La réduction deux ne concerne que les K-nœuds.

Réduction polygone-à chaine. La réduction polygone-à chaine considère en général que le polygone soit indépendant du graphe par le fait qu'il soit relié au reste du graphe que par ses K-nœuds. Autrement dit, s'il se trouve qu'il soit lié par des nœuds autres que les cas K-nœuds et comme ces nœuds sont imparfaits une solution est alors à envisagée. Theologou *et* Carlier (1991) ont proposé la construction suivante :
Considérons un lien $e = (u, v)$, où *u* et v sont des nœuds imparfaits de probabilités respectives p_u et p_v. Nous constatons que le lien *l* fonctionne avec une probabilité p_l lorsque *e*, *u* and *v* fonctionnent avec une probabilité p_i $(i = e, u, v)$ et est telle que :

$$p_l = p_e p_u p_v \tag{7.13}$$

Le fait de pivoter sur le lien *l* en appliquant le théorème de factorisation, entrainera d'un coté la contraction de l'arrête *e* et le fusionnement des nœuds *u* et *v* pour former un nouveau nœud parfait. Par contre si le lien *l* arrête de fonctionner, la cause de cet arrêt ne peut être connue a priori du fait qu'il se pourrait que ce soit l'arrête ou les deux nœuds en même temps qui soient défaillants. De toute façon et dans tous les cas, ce lien sera perdu et donc supprimé.

Après cette opération de pivotage, les fiabilités de *u* et de *v* auront de nouvelles valeurs et leurs expressions sont comme suit :

$$p'_v = \Pr\ v \text{ fonctionne } \ v \text{ ou } e \text{ ou } u \text{ en pannes}) = \frac{p_v(q_u + p_u q_e)}{q_v + p_v q_u + p_v p_u q_e} \tag{7.14}$$

$$p'_u = \Pr\ u \text{ fonctionne } \ u \text{ ou } e \text{ ou } v \text{ en pannes}) = \frac{p_u(q_v + p_v q_e)}{q_u + p_u q_v + p_u p_v q_e} \tag{7.15}$$

Nous pouvons remarquer que ces deux derrières équations montrent clairement que *u* et *v* sont intimement liés. Pour éviter une telle situation compte-tenu des conditions de départ, Théologou *et* Carlier ont apporté une nouvelle construction de *u* et de *v*. L'idée est comme suit :

$$R(G_k) = \prod_{v \in K} p_v R(G'_k) \tag{7.16}$$

Supposons que les K-nœuds sont tous connectés et parfaits, autrement dit si ce n'est pas le cas la fiabilité du réseau original s'écrit : où G'_k et G_k sont construits avec des K-nœuds parfaits.

Il existe alors au moins deux K-nœuds dans le graphe et qu'il est toujours possible de trouver une arête dans le graphe avec une extrémité parfaite. Soit *e* une telle arête d'extrémités *u* parfait et *v* imparfait. Le fait de remplacer dans (14) $p_u = 1$ et comme $p_v \neq 1$, on obtient :

$$p'v = \frac{p_V q_e}{q_V + p_V q_e} \tag{7.17}$$

Nous remarquons que dans l'état actuel, la défaillance de v ne dépend que de l'arrête e et donc, u et v sont rendus indépendants et par conséquent avec une telle construction tous les nœuds sont forcés d'être indépendants.

Comme le nœud v reste dans le graphe ou le lien l est défaillant, alors les autres liens qui ont pour l'une de leurs extrémités v peuvent être factorisés et que la défaillance de v dépendra alors de la défaillance de ces arêtes. Par conséquent, il s'agit de construire la formulation générale qui permet de représenter la fiabilité relative au nœud imparfait v à une étape quelconque de la factorisation lorsque les arêtes $(e_1, e_2, \cdots, e_r)$ incidentes à v sont factorisées. Cette fiabilité est donnée par (18) après l'application du théorème de factorisation:

$$
\begin{aligned}
p''_v &= \Pr\ v \text{ fonctionne } \ (v \text{ ou } e_1 \text{ en panne}) \wedge \cdots \wedge (v \text{ ou } e_r \text{ en panne}) \\
&= \frac{p_v \quad {}_{j=1}^{r} q_{e_j}}{q_v + p_v \quad {}_{j=1}^{r} q_{e_j}}
\end{aligned}
\tag{7.18}
$$

7.7 Réduction polygone-à chaine de type 1

Nous énonçons à présent le théorème 4 qui permet d'avancer les résultats de la réduction polygone-à chaine de type 1 dans le cadre des graphes qui admettent une la structure d'un polygone de type 1.

Théorème 7.4.

Supposons qu'un graphe G contienne un polygone de type 1 et soit G' le graphe obtenu par transformation du graphe G par le remplacement des arêtes e_1 *et* e_2 de fiabilités originales p_1 *et* p_2 par e'_1 *et* e'_2 de fiabilités p'_1 *et* p'_2 et les nœuds *a* et *b* de fiabilités originales p_a *et* p_b par les fiabilités p'_a *et* p'_b (figure 6, ci-dessus) et soit Ω un facteur de multiplication. Alors : $R\ \ G\ \ = \Omega R(G')$

Avec $\Omega = \ \ \delta + A\ \ (\delta + B)/\delta$, $p'_1 = \delta/\ \ \delta + AC$, $p'_2 = \delta/\ \ \delta + BD$, $p'_a = (\delta + AC)/\ \ \delta + A$, $p'_b = (\delta + BD/\ \ \delta + B$ et :

$$\delta = p_a p_b \; p_1 p_2 + \; p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3$$
$$A = p_2 p_b \; 1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a$$
$$B = p_1 p_a \; 1 - p_2 p_b - \; p_3 p_b + p_2 p_3 p_b$$
$$C = \frac{p_a q_1 q_3}{q_a + p_a q_1 q_3}$$
$$D = \frac{p_b q_2 q_3}{q_b + p_b q_2 q_3}$$

Preuve :

En appliquant le théorème de factorisation sur le lien *(s,*e_1, a), on obtient les graphes réduits $(G * e_1)$ et $(G - e_1)$ (voir figure 7.7). L'expression de la fiabilité de graphe *G* devient :

$$R \;\; G \;\; = p_1 p_a R \;\; G * e_1 \;\; + \;\; 1 - p_1 p_a \;\; R \;\; G - e_1 \qquad (7.19)$$

Par application des résultats introduits par Theologou and Carlier (1991), et comme le nœud *a* est un nœud imparfait, il devient lors de la décomposition un nœud pivot. Par conséquent la fiabilité du nœud *a* dans le graphe qui correspond à $G - e_1$, est modifiée et sera remplacée par la relation suivante :

$$p_a' = \frac{p_a q_1}{q_a + p_a q_1} \qquad (7.20)$$

En décomposant sur le lien *(s,*e_2, b), des graphes réduits $G * e_1$ et $G - e_1$. L'expression de la fiabilité de graphe *G* devient :

$$\begin{aligned} R(G) = p_1 p_a \;\; p_2 p_b R \;\; (G * e_1 \;\; * e_2 \;\; + \;\; 1 - p_2 p_b \;\; R(\; G * e_1 \;\; - e_2)] \\ + \;\; 1 - p_1 p_a \;\; [p_2 p_b R \quad G - e_1 \;\; * e_2 \;\;] \end{aligned} \qquad (7.21)$$

De la même manière que le nœud a, la fiabilité du nœud b dans le graphe correspondant à $G * e_1 \;\; - e_2$ est modifiée et sera égale a :

$$p'_b = \frac{p_b q_2}{q_b + p_b q_2} \qquad (7.22)$$

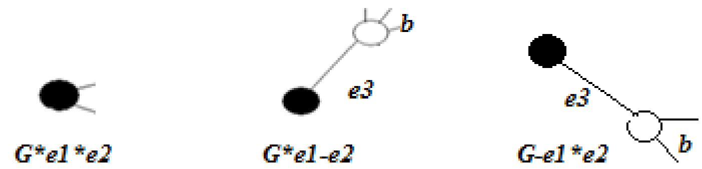


Figure 7.11. Graphes générés par la décomposition sur le lien *(s,*e_2, b).

En décomposant maintenant sur le lien *(s,*e_3, b) du graphe réduit $G * e_1 \;\; - e_2$ et sur le lien *(s,* e_3, a) du graphe réduit $G - e_1 \;\; * e_2$ (voir figure 8), on obtient finalement :

$$R(G) = p_1 p_a\left[p_2 p_b R((G*e_1)*e_2) + (1-p_2 p_b)\begin{bmatrix} p_3 p'_b R(((G*e_1)-e_2)*e_3) + \\ (1-p_3 p'_b)R(((G*e_1)-e_2)-e_3) \end{bmatrix}\right] + \\ (1-p_1 p_a)[p_2 p_b[p_3 p'_a R(((G-e_1)*e_2)*e_3) + (1-p_3 p'_a)R(((G-e_1)*e_2)-e_3)]] \tag{7.23}$$

Dans le graphe $G - e_1 \;\; * e_2 \;\; - e_3$ la nouvelle fiabilité du nœud est donnée par :

$$p''_a = \frac{p_a q_1 q_3}{q_a + p_a q_1 q_3} \tag{7.24}$$

Avec le graphe $G * e_1 \;\; - e_2 \;\; - e_3$, la nouvelle fiabilité du nœud *b* est donnée par :

$$p''_b = \frac{p_b q_2 q_3}{q_b + p_b q_2 q_3} \tag{7.25}$$

Comme on peut le remarquer, plusieurs états peuvent induire un graphe identique. On peut donc établir l'égalité suivante :

$$R\;(G * e_1 \;\; * e_2) = R\;(\;G - e_1 \;\; * e_2 \;\; * e_3 = R(\;\; G * e_1 \;\; - e_2 \;\; * e_3) \tag{7.26}$$

En groupant et en éliminant des termes, l'expression de la fiabilité du graphe devient :

$$\begin{aligned} R(G) &= p_a p_b[p_1 p_2 + p_1 p_3 + p_2 p_3 - 2p_1 p_2 p_3]R((G*e_1)*e_2) \\ &+ p_1 p_a[1 - p_2 p_b - p_3 p_b + p_2 p_3 p_b]R(((G*e_1)-e_2)-e_3) \\ &+ p_2 p_b[1 - p_1 p_a - p_3 p_a - p_1 p_3 p_a]R(((G-e_1)*e_2)-e_3) \end{aligned} \tag{7.27}$$

Procédons maintenant sur le graphe G'. L'application du théorème de factorisation sur le lien (s, e'_1, a) du graphe *G'*, conduit à :

$$R\;\;G' \;\; = p'_1 p'_a R\;\;G' * e'_1 \;\; + \;\; 1 - p'_1 p'_a \;\; R(G' - e'_1) \tag{7.28}$$

La fiabilité du nœud *a* dans le graphe correspondant à $G' - e'_1$ a été modifiée selon (29) :

$$(p'_a)' = \frac{p'_a q'_1}{q'_a + p'_a q'_1} \tag{7.29}$$

Finalement l'expression de la fiabilité du graphe *G'* est obtenue comme suit :

$$\begin{aligned} R(G') = {} & p'_1 p'_a \left[p'_2 p'_b R((G' * e'_1) * e'_2) + (1 - p'_2 p'_b) R((G' * e'_1) - e'_2)\right] \\ & + (1 - p'_1 p'_a)\left[p'_2 p'_b R((G' - e'_1) * e'_2)\right] \end{aligned} \tag{7.30}$$

Par contre la nouvelle expression de la fiabilité relative au nœud b devient :

$$(p'_b)' = \frac{p'_b q'_2}{q'_b + p'_b q'_2} \tag{7.31}$$

Comme les graphe *G* and *G'* sont identiques, nous obtenons :

$$R\left(\left(G * e_1\right) * e_2\right) = R\left(\left(G' * e'_1\right) * e'_2\right) \quad \text{et} \quad R\left(\left(\left(G * e_1\right) - e_2\right) - e_3\right) = R\left(\left(G' * e'_1\right) - e'_2\right)$$

Cette dernière expression est valide si et seulement si :

$$p''_a = (p'_a)'$$

et l'égalité $R\left(\left(\left(G - e_1\right) * e_2\right) - e_3\right) = R\left(\left(G' - e'_1\right) * e'_2\right)$ est valide si et seulement si :

$$p''_b = (p'_b)'$$

Finalement, suite à la relation d'équivalence $R(G) = \Omega R(G')$, et la dernière formule, nous déduisons le système suivant :

$$\begin{cases} p_a p_b \left(p_1 p_2 + p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3\right) = \Omega\, p'_1 p'_2 p'_a p'_b \\ p_1 p_a \left(1 - p_2 p_b - p_1 p_3 + p_3 p_b + p_2 p_3 p_b\right) = \Omega\, p'_1 p'_a \left(1 - p'_2 p'_b\right) \\ p_2 p_b \left(1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a\right) = \Omega\, p'_1 p'_b \left(1 - p'_2 p'_a\right) \\ \dfrac{p_a q_1 q_3}{q_a + p_a q_1 q_3} = \dfrac{p'_a q'_1}{q'_a + p'_a q'_1} \\ \dfrac{p_b q_2 q_3}{q_b + p_b q_2 q_3} = \dfrac{p'_b q'_2}{q'_b + p'_b q'_2} \end{cases} \tag{7.32}$$

La résolution du système (7.32) nous donne les expressions des paramètres $\Omega, p'_1, p'_2, p'_a,$ et $p'_{b_1},$ et qui se résument comme suit :

$$
\begin{aligned}
\Omega &= \frac{\delta + \mathrm{A} \quad \delta + \mathrm{B}}{\delta} \\
\mathrm{p'}_1 &= \frac{\delta}{(\delta + \mathrm{A\,C})} \\
\mathrm{p'}_2 &= \frac{\delta}{(\delta + \mathrm{B\,D})} \\
\mathrm{p'}_\mathrm{a} &= \frac{(\delta + \mathrm{A\,C})}{(\delta + \mathrm{A})} \\
\mathrm{p'}_\mathrm{b} &= \frac{(\delta + \mathrm{B\,D})}{(\delta + \mathrm{B})}
\end{aligned}
\tag{7.33}
$$

avec δ, A, B, C et D tels que :

$$
\begin{aligned}
\delta &= p_a p_b \;\; p_1 p_2 + \; p_1 p_3 + p_2 p_3 - 2 p_1 p_2 p_3 \\
A &= p_2 p_b \;\; 1 - p_1 p_a - p_3 p_a + p_1 p_3 p_a \\
B &= p_1 p_a \;\; 1 - p_2 p_b - \; p_3 p_b + p_2 p_3 p_b \\
C &= \frac{p_a q_1 q_3}{q_a + p_a q_1 q_3} \\
D &= \frac{p_b q_2 q_3}{q_b + p_b q_2 q_3}
\end{aligned}
\tag{7.34}
$$

Nous venons de démontrer le théorème 7.4, passons maintenant au cas de la réduction polygone-à chaine de type 7.

7.8 Réduction polygone-à chaine de type 7

Théorème 7.5.

Supposons qu'un graphe G contienne un polygone de type 7 et soit G' le graphe obtenu par transformation du graphe G par le remplacement des arêtes e_1, e_2, e_3, e_4, e_5 et e_6 de fiabilités originales p_1, p_2, p_3, p_4, p_5 et p_6 par e_1' *et* e_2' de fiabilités p_1' *et* p_2' et les nœuds *a* et *b* de fiabilités originales p_a *et* p_b par les fiabilités p_a' *et* p_b' (figure 7.11, ci-dessous) et soit Ω un facteur de multiplication. Alors $R \;\; G \;\; = \Omega R(G')$ et,

$$
\begin{cases}
\gamma = p_1p_2p_3p_4p_5p_6p_ap_b[1+\frac{q_1}{p_1}+\frac{q_2}{p_2}+\frac{q_3}{p_3}+\frac{q_4}{p_4}+\frac{q_5}{p_5}+\frac{q_6}{p_6}] \\
B = p_1p_2p_3p_4p_5p_6p_ap_b[\frac{q_3q_4}{p_3p_4}+\frac{q_2q_5}{p_2p_5}+\frac{q_2q_4}{p_2p_4}+\frac{q_1q_5}{p_1p_5}+\frac{q_1q_6}{p_1p_6}+\frac{q_3q_5}{p_3p_5}+\frac{q_2q_6}{p_2p_6}] \\
A = p_1p_2p_3p_5p_6p_b[1-p_1p_a-p_4p_a+p_1p_4p_a] \\
\delta = p_1p_2p_4p_5p_a[1-p_3p_b-p_6p_b+p_3p_6p_b] \\
C = \frac{p_aq_1q_4}{(q_a+p_aq_1q_4)} \\
D = \frac{p_bq_3q_6}{(q_b+p_bq_3q_6)}
\end{cases}
\tag{7.36}
$$

$$
\Omega = \frac{(\gamma+A)(\gamma+B)(\delta+\gamma)}{\gamma^2} \quad p_1' = \frac{(\gamma)}{(\gamma+AC)} \quad p_2' = \frac{(\gamma)}{(\gamma+B)} \quad p_3' = \frac{(\gamma)}{(\gamma+D\delta)} \quad p_a' = \frac{(\gamma+AC)}{(\gamma+A)} \tag{7.37}
$$

$$
p_b' = \frac{(\gamma+D\delta)}{(\gamma+\delta)}
$$

Preuve :

Considérons à présent un graphe contenant un polygone de type 7 comme montré à la figure 7.11(G) et dont les nœuds *a* et *b* peuvent être sujets à des défaillances avec des fiabilités de bon fonctionnement respectives p_a et p_b.

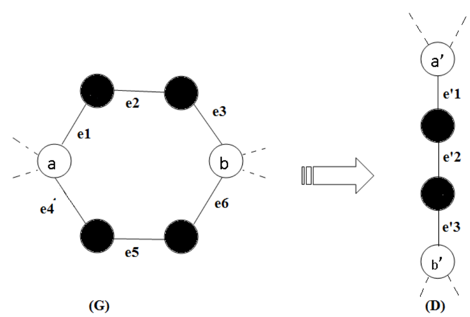


Figure 7.12. Réduction polygone-à chaine de type 7

En pivotant successivement sur e_1, e_2, e_3, e_4, e_5 et e_6 on obtient les graphes reportés dans la figure 7.12 et la table 7.4 :

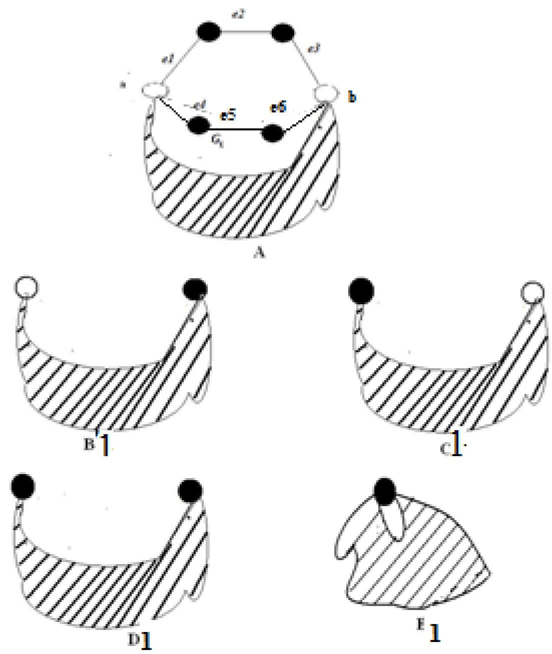


Figure 7.13. Graphes non-défaillants induits suite à la réduction polygone-à chaine de type 7.

Table 7.4. États de non-défaillants et les probabilités des graphes ainsi déduits.

G_A	États	Probabilités
B1	$\overline{F_a F_1} F_2 F_3 \overline{F_a F_4} F_5 F_6$	$\alpha = p_2 p_3 p_5 p_6 p_b[(1-p_1 p_a)(1-p_4 p_a)]$ $= p_2 p_3 p_5 p_6 p_b[1-p_1 p_a - p_4 p_a + p_1 p_4 p_a]$
C1	$F_1 F_2 \overline{F_b F_3} F_4 F_5 \overline{F_b F_6}$	$\delta = p_1 p_2 p_4 p_5 p_a[(1-p_3 p_b)(1-p_6 p_b)]$ $= p_1 p_2 p_4 p_5 p_a[1-p_3 p_b - p_6 p_b + p_3 p_6 p_b]$
D1	$F_a F_1 \overline{F_2} F_3 \overline{F_4} F_5 F_6 F_b ; F_a F_1 \overline{F_2} F_3 F_4 \overline{F_5} F_6 F_b ;$ $F_a F_1 F_2 \overline{F_3}\, \overline{F_4} F_5 F_6 F_b ; F_a F_1 F_2 \overline{F_3} F_4 \overline{F_5} F_6 F_b ;$ $F_a F_1 \overline{F_2} F_3 F_4 F_5 \overline{F_6} F_b ; F_a \overline{F_1} F_2 F_3 F_4 \overline{F_5} F_6 F_b ;$ $F_a \overline{F_1} F_2 F_3 F_4 F_5 \overline{F_6} F_b$	$\beta = (p_1 q_2 p_3 q_4 p_5 p_6 + p_1 q_2 p_3 p_4 q_5 p_6 + p_1 p_2 q_3 q_4 p_5 p_6$ $+ p_1 p_2 q_3 p_4 q_5 p_6 + p_1 q_2 p_3 p_4 p_5 q_6 + q_1 p_2 p_3 p_4 q_5 p_6 +$ $q_1 p_2 p_3 p_4 p_5 q_6) p_a p_b$ $= p_1 p_2 p_3 p_4 p_5 p_6 p_a p_b[\frac{q_2 q_4}{p_2 p_4} + \frac{q_2 q_5}{p_2 p_5} + \frac{q_3 q_4}{p_3 p_4} + \frac{q_3 q_5}{p_3 p_5} +$ $\frac{q_2 q_6}{p_2 p_6} + \frac{q_1 q_5}{p_1 p_5} + \frac{q_1 q_6}{p_1 p_6}]$
E1	$F_a F_1 F_2 F_3 F_4 F_5 F_6 F_b ; \overline{F_a F_1} F_2 F_3 F_4 F_5 F_6 F_b ;$ $F_a F_1 \overline{F_2} F_3 F_4 F_5 F_6 F_b ; F_a F_1 F_2 \overline{F_3} F_4 F_5 F_6 F_b ;$ $F_a F_1 F_2 F_3 \overline{F_4} F_5 F_6 F_b ; F_a F_1 F_2 F_3 F_4 \overline{F_5} F_6 F_b$	$\gamma = (p_1 p_2 p_3 p_4 p_5 p_6 + q_1 p_2 p_3 p_4 p_5 p_6 + p_1 q_2 p_3 p_4 p_5 p_6 +$ $p_1 p_2 q_3 p_4 p_5 p_6 + p_1 p_2 p_3 q_4 p_5 p_6 + p_1 p_2 p_3 p_4 q_5 p_6 +$ $p_1 p_2 p_3 p_4 p_5 q_6) p_a p_b$ $= p_1 p_2 p_3 p_4 p_5 p_6 p_a p_b (1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4} + \frac{q_5}{p_5} + \frac{q_6}{p_6})$

Comme la décomposition s'effectue sur e_1 et sur e_4 sur le graphe réduit B1 et sur e_3 et e_6 sur le graphe réduit C1, il s'en vient que les nouvelles valeurs des fiabilités des nœuds a et b donnée par l'équation (20) et (21), sont :

$$p''_a = \frac{p_a q_1 q_4}{q_a + p_a q_1 q_4}$$

$$p''_b = \frac{p_b q_3 q_6}{q_b + p_b q_3 q_6}$$

Reprenons l'application du théorème de factorisation sur le graphe réduit *G'* de *G* (polygone de type 7) et soient $e_1, e_2, et\, e_3$ les arêtes de *G'* de fiabilités respectives $p_1, p_2, et\, p_3$. La table 7.5 nous donne les graphes induits suite à la décomposition de *G'*, les états et les probabilités relatives.

Table 7.5. États non-défaillants et les probabilités des graphes induits par le processus de factorisation du graphe $G'_{K'}$.

Graphe	States	Probabilités
B'	$\overline{F_a F_1} F_2 F_3 F_b$	$\alpha' = (1 - p_1^{'} p_a^{'}) p_2^{'} p_3^{'} p_b^{'}$
C'	$F_a F_1 F_2 \overline{F_b F_3}$	$\beta' = (1 - p_3^{'} p_b^{'}) p_1^{'} p_2^{'} p_a^{'}$
D'	$F_a F_1 \overline{F_2} F_3 F_b$	$\delta' = p_1^{'} q_2^{'} p_3^{'} p_a^{'} p_b^{'}$
E'	$F_a F_1 F_2 F_3 F_b$	$\gamma' = p_1^{'} p_2^{'} p_3^{'} p_a^{'} p_b^{'}$

Les graphes suivants (figure 7.13), expliquent les formules des états et les probabilités correspondantes de la table 7.5. Notez que pour le graphe B' et C' respectivement les probabilités conditionnelles sur le nœud *a* et l'arête e'_1 (graphe B') et l'arête e'_3 et le nœud *b* (graphe B') conduisent aux relations sur les nouvelles valeurs des fiabilités sur les nœuds a et b.

$$p''_a = (p'_a)' = \frac{p'_a\, q'_1}{q'_a + p'_a\, q'_1}$$

$$p''_b = (p'_b)' = \frac{p'_b\, q'_3}{q'_b + p'_b\, q'_3}$$

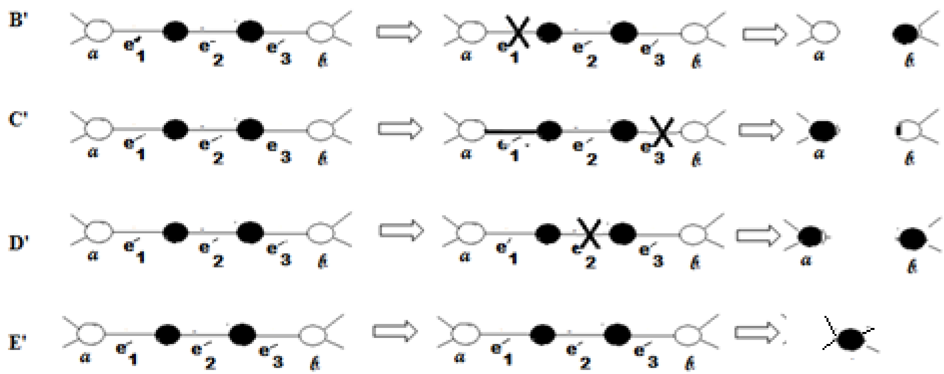


Figure 7.14. Transformation du graphe réduit *G'* par application de la factorisation

Utilisons présentement la relation $R(G_K) = \Omega R(G'_{K'})$, nous pouvons alors identifier les coefficients α, β, δ et γ comme suit :

$$A.R(G_{B,K_B}) + B..R(G_{C,K_C}) + \delta.R(G_{D,K_D}) + \gamma.R(G_{E,K_E}) = \Omega.[(1 - p'_1 p'_a) p'_2 p'_3 p'_b R(G'_{B',K'_{B'}})$$
$$+ (1 - p'_3 p'_b) p'_1 p'_2 p'_a R(G'_{C',K'_{C'}}) + p'_1 q'_2 p'_3 p'_a p'_b R(G'_{D',K'_{D'}}) + p'_1 p'_2 p'_3 p'_a p'_b R(G'_{E',K'_{E'}})]$$

et :
$$\frac{p_a q_1 q_4}{q_a + p_a q_1 q_4} = \frac{p'_a q'_1}{q'_a + p'_a q'_1}$$
$$\frac{p_b q_3 q_6}{q_b + p_b q_3 q_6} = \frac{p'_b q'_3}{q'_b + p'_b q'_3}$$

Égalisons les termes des deux cotés de l'équation $R(G_K) = \Omega R(G'_{K'})$, il s'ensuit l'expression des égalités suivantes :

$$\begin{cases} p_1 p_2 p_3 p_4 p_5 p_6 p_a p_b [1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4} + \frac{q_5}{p_5} + \frac{q_6}{p_6}] = \Omega [p'_1\, p'_2\, p'_3\, p'_a\, p'_b] \\ p_1 p_2 p_3 p_4 p_5 p_6 p_a p_b [\frac{q_3 q_4}{p_3 p_4} + \frac{q_2 q_5}{p_2 p_5} + \frac{q_2 q_4}{p_2 p_4} + \frac{q_1 q_5}{p_1 p_5} + \frac{q_1 q_6}{p_1 p_6} + \frac{q_3 q_5}{p_3 p_5} + \frac{q_2 q_6}{p_2 p_6}] = \Omega [p'_1\, q'_2\, p'_3\, p'_a\, p'_b] \\ p_1 p_2 p_4 p_5 p_a [1 - p_3 p_b - p_6 p_b + p_3 p_6 p_b] = \Omega [p'_1\, p'_2\, p'_a (1 - p'_3\, p'_b)] \\ p_1 p_2 p_3 p_5 p_6 p_b [1 - p_1 p_a - p_4 p_a + p_1 p_4 p_a] = \Omega [p'_2\, p'_3\, p'_b (1 - p'_1\, p'_a)] \\ \frac{p_a q_1 q_4}{q_a + p_a q_1 q_4} = \frac{p'_a q'_1}{q'_a + p'_a q'_1} \\ \frac{p_b q_3 q_6}{q_b + p_b q_3 q_6} = \frac{p'_b q'_3}{q'_b + p'_b q'_3} \end{cases} \tag{7.35}$$

En résolvant le système on obtient les relations (7.36) et (7.37) suivantes :

$$\begin{cases} \gamma = p_1p_2p_3p_4p_5p_6p_ap_b[1+\frac{q_1}{p_1}+\frac{q_2}{p_2}+\frac{q_3}{p_3}+\frac{q_4}{p_4}+\frac{q_5}{p_5}+\frac{q_6}{p_6}] \\ B = p_1p_2p_3p_4p_5p_6p_ap_b[\frac{q_3q_4}{p_3p_4}+\frac{q_2q_5}{p_2p_5}+\frac{q_2q_4}{p_2p_4}+\frac{q_1q_5}{p_1p_5}+\frac{q_1q_6}{p_1p_6}+\frac{q_3q_5}{p_3p_5}+\frac{q_2q_6}{p_2p_6}] \\ A = p_1p_2p_3p_5p_6p_b[1-p_1p_a-p_4p_a+p_1p_4p_a] \\ \delta = p_1p_2p_4p_5p_a[1-p_3p_b-p_6p_b+p_3p_6p_b] \\ C = \frac{p_aq_1q_4}{(q_a+p_aq_1q_4)} \\ D = \frac{p_bq_3q_6}{(q_b+p_bq_3q_6)} \end{cases} \tag{7.36}$$

$$\Omega = \frac{(\gamma+A)(\gamma+B)(\delta+\gamma)}{\gamma^2} \quad p_1^{'} = \frac{(\gamma)}{(\gamma+AC)} \quad p_2^{'} = \frac{(\gamma)}{(\gamma+B)} \quad p_3^{'} = \frac{(\gamma)}{(\gamma+D\delta)} \quad p_a^{'} = \frac{(\gamma+AC)}{(\gamma+A)} \tag{7.37}$$

$$p_b^{'} = \frac{(\gamma+D\delta)}{(\gamma+\delta)}$$

Algorithme

Début

Lecture des données d'entrées:

G : le graphe connexe non divisible (1 seule composante connexe)

$P_{i,j} = (E_i, E_j)$: Matrice associée construite à partir des arêtes du graphe

$K \subseteq V, |K| \geq 2$

$M = 1$

Tant que qu'il existe une réduction simple des 3 types suivants :

Appel procédures :

- Réduction Séries : $P_i = P_j \times P_k$, Supprimer le lien (E_i, E_j) et mettre à jour le vecteur des probabilité et la matrice associée au graphe.
- Réduction parallèle : $P_i = 1 - (1 - P_j) \times (1 - P_k)$; supprimer l'un des liens et mettre à jour la vecteur des probabilités et la matrice associée au graphe.
- Réduction degré 2. Soit $M = M \times \Omega$ pour chaque réduction.

Fin Tant que.

Commencer par explorer les structures complexes.

Soit S un tel sommet de départ.

Déterminer les sommets ascendants (et les sommets ascendants de deuxième niveau et de niveaux plus bas, selon le type de polygone)

Tant qu'un polygone existe faire

Si un polygone de type T existe appliquez la réduction de ce type. On commence toujours par chercher celui de type 1, puis de type 2, etc.

- Mémoriser dans une pile les liens à supprimer.
- Supprimer les liens dans la matrice associée au graphe.
- Reconstruire les nouvelles chaines à partir des liens mémorisés en tissant les liens dans la matrice relative au graphe.
- Mettre à jour le vecteur des probabilités avec les nouvelles valeurs relatives à chaque type T.
- Appliquer les réductions série, parallèle et de degré 2.
- Mettre à jour le vecteur des probabilités.

- Mettre à jour la matrice relative au graphe.

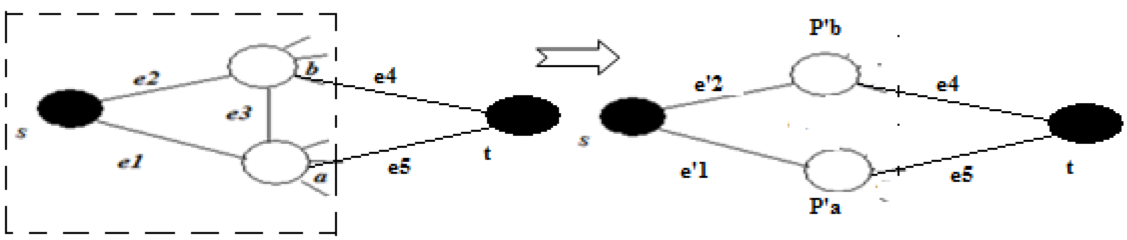
Fin-Si
Fin tant que
Construire le graphe à partir de la matrice résultante.
- ***Si*** c'est une matrice simple qui mettez en relief une chaine
  Calculer la fiabilité comme le produit des probabilités des liens
  Sinon
  Imprimer matrice n'est plus décomposable
  Appliquer un algorithme quelconque qui calcule la fiabilité (SDP, BDD, …)

Fin-si
Imprimez la fiabilité
Fin algorithme

7.8.1 Exemples d'application

Exemple 7.1.

Reprenons l'exemple de la figure 7.6 tout en supposant que tous les nœuds sont imparfaits. Considérons que les valeurs de bon fonctionnement de chaque nœuds et de chaque lien est égal à 0.9. Le graphe suivant nous donne les étapes de calcul.



$$G = \begin{matrix} e_1 & e_2 & \\ & e_3 & e_5 \\ & & e_4 \end{matrix} \; ; \quad P-edge = (p_1, p_2, p_3, p_4, p_5) \; ; P-Nodes = (1, p_a, p_b, 1)$$

Etape 1:

Node_inital = S; Pile = Node_inital

Liste-succ(S) = (a, b)

Pile = Pile ∪ Liste-succ(S)   (donc Pile = (S, a, b))

X = dépiler(Pile) = (b)

Y = Liste-succ(X) = (a, t)

Si Y ≠ 1_sommet_Terminal

Z = X ∩ Pile = (b) ∩ (S, a) = (a)

Donc Polygone de type 1 a été trouvé

Mise-à-jour des vecteurs et de la matrice

$$G = \begin{bmatrix} \square & e_1 & e_2 & \square \\ \square & \square & \square & e_5 \\ \square & \square & \square & e_4 \\ \square & \square & \square & \square \end{bmatrix} ; P - edge = (p_1 = \frac{\delta}{\delta + A.C}, p_2 = \frac{\delta}{\delta + B.D}, 0, p_4, p_5) ;$$

$$P - Nodes = (1, p_a = \frac{\delta + A.C}{\delta + A}, p_b = \frac{\delta + B.D}{\delta + B}, 1)$$

X = dépiler(Pile) = (a)

Y = Liste-succ(X) = (t)

Si Y ≠ 1_sommet_Terminal

NON
Prob = Prob × Prob(X, t) = 1 × p_5 = p_5
Pile = dépiler(Pile) = (S)
Prob = Prob × Prob(Pile, X) = $p_5 \times p_1 \times p_a$

Mise-à-jour des vecteurs et de la matrice

$$e'_1 = (S, t)$$

$$G = \begin{bmatrix} \square & \square & e_2 & e'_1 \\ \square & \square & \square & \square \\ \square & \square & \square & e_4 \\ \square & \square & \square & \square \end{bmatrix} ; P - edge = (p_1 = Prob, p_2 = \frac{\delta}{\delta + B.D}, 0, p_4, 0) ;$$

Prob = 1

Liste-succ(S) = (b)

Pile = Pile ∪ Liste-succ(S)   (donc Pile = (S, b))

X = dépiler(Pile) = (b)

Y = Liste-succ(X) = (t)

Si Y ≠ 1_sommet_Terminal

NON
Prob = Prob × Prob(X, t) = 1 × p_4 = p_4
Pile = dépiler(Pile) = (S)
Prob = Prob × Prob(Pile, X) = $p_4 \times p_2 \times p_b$

Mise-à-jour des vecteurs et de la matrice

$e'_2 = (S, t)$

$$G = \begin{bmatrix} \square & \square & \square & e'_1 || e'_2 \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{bmatrix} ; P - edge = (p_1 = \ Prob, p_2 = \ Prob, 0{,}0{,}0) ;$$

$P - edge = (p''_1 = \ 1 - (1 - q_1) \times (1 - q_1), 0{,}0{,}0{,}0)$

Fin-si

Fin Algorithm

Calcul :

```
function imperfect()
  p1 = .9
  p2 = .9
  p3 =.9
  pa = .9
  pb = .9
  A = p2*pb*(1-p1*pa-p3*pa+p1*p3*pa)
  B = p1*pa*(1-p2*pb-p3*pb+p2*p3*pb)
  delta = pa*pb*(p1*p2+p1*p3+p2*p3-2*p1*p3*pa)
  C=(pa*(1-p1)*(1-p3))/((1-pa) +pa*(1-p1)*(1-p3))
  D=(pb*(1-p2)*(1-p3))/((1-pb) +pb*(1-p2)*(1-p3))
  omega = (delta + A)*( delta + B)/ delta
  p1p = delta /( delta + A*C)
  p2p = delta /( delta + B*D)
  pap = (delta + A*C)/( delta + A)
  pbp = (delta + B*D)/( delta + B)
  x = 1-(1-p1p*pbp*.9)*(1-p2p*pap*.9)
  y = x*.81*omega
end
```

A = B = 0.08829; eta = 0.78732; C = D = 0.0825688073394495; p1p = p2p = 0.990825688073394; pap = pbp = 0.907493061979649; omega = 0.9738008333333333.

Comme le graphe s'est transformé en deux chaines parallèles. La fiabilité est :

R = 1 – (1 - p2p * pbp * p4)*(1 - p1p * pap * p4) = 0.963614702184995

R(G) = R * omega * Ps * Pt = 0.963614702184995 * 0.9738008333333333 * 0.9 * 0.9 * = 0.760078728

Exemple 7.2.

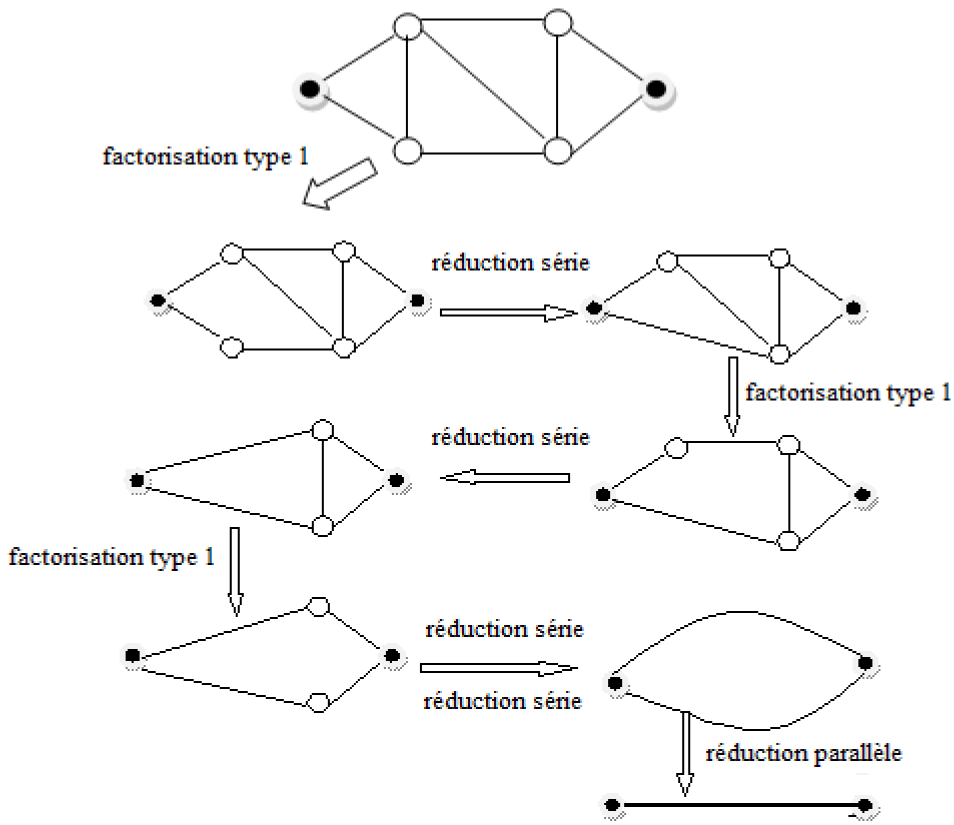
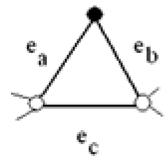
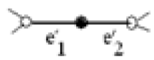
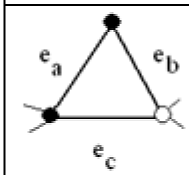
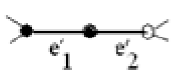
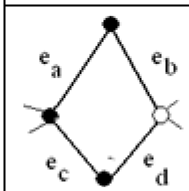
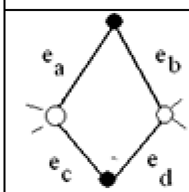
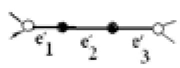
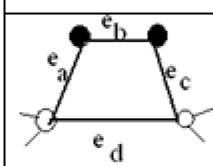
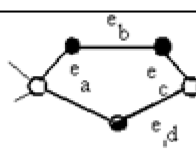
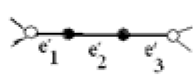
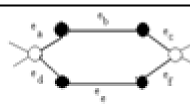
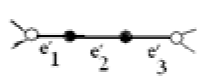


Figure 7.15. Réduction polygone-à chaine de type 1, série et parallèle

Table 7.6. Réduction polygone-à chaine de type 1 à 7 pour les réseaux imparfaits

		$\delta = p_a p_b[p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3]$ $A = p_2p_b[1 - p_1p_a - p_3p_a + p_1p_3p_a]$ $B = p_1p_a[1 - p_2p_b - p_3p_b + p_2p_3p_b]$ $C = \frac{p_aq_1q_3}{(q_a + p_aq_1q_3)}$ $D = \frac{p_bq_2q_3}{(q_b + p_bq_2q_3)}$	$\Omega = \frac{(\delta+A)(\delta+B)}{\delta}$; $p'_1 = \frac{(\delta)}{(\delta+AC)}$ $p'_2 = \frac{(\delta)}{(\delta+BD)}$ $p'_a = \frac{(\delta+AC)}{(\delta+A)}$ $p'_b = \frac{(\delta+BD)}{(\delta+B)}$
		$\delta = p_a p_b[p_1p_2 + p_1p_3 + p_2p_3 - 2p_1p_2p_3]$ $A = p_2p_b[1 - p_1p_a - p_3p_a + p_1p_3p_a]$ $B = p_1p_a[1 - p_2p_b - p_3p_b + p_2p_3p_b]$ $C = \frac{p_aq_1q_3}{(q_a + p_aq_1q_3)}$ $D = \frac{p_bq_2q_3}{(q_b + p_bq_2q_3)}$	$\Omega = \frac{[\delta+A][\delta+B]}{\delta}$ $p'_1 = \frac{[\delta]}{[\delta+AC]}$ $p'_2 = \frac{(\delta)}{(\delta+BD)}$ $p'_a = \frac{[\delta+AC]}{[\delta+A]}$ $p'_b = \frac{[\delta+BD]}{[\delta+B]}$
		$\delta = p_1p_2p_3p_4p_a[1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4}]$ $A = p_1p_2p_3p_4p_a[1 + \frac{q_1q_4}{p_1p_4} + \frac{q_2q_3}{p_2p_3} + \frac{q_1q_3}{p_1p_3}]$ $B = p_1p_3[1 - p_2p_a - p_4p_a + p_2p_4p_a]$ $C = \frac{p_aq_2q_4}{(q_a + p_aq_2q_4)}$	$\Omega = \frac{[\delta+A][\delta+B]}{\delta}$ $p'_1 = \frac{(\delta)}{(\delta+AC)}$ $p'_2 = \frac{(\delta)}{(\delta+BC)}$ $p'_a = \frac{[\delta+BC]}{[\delta+B]}$
		$\gamma = p_1p_2p_3p_4p_ap_b[1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4}]$ $B = p_1p_2p_3p_4p_ap_b[\frac{q_1q_4}{p_1p_4} + \frac{q_2q_3}{p_2p_3}]$ $A = p_2p_4p_b[1 - p_1p_a - p_3p_a + p_1p_3p_a]$ $\delta = p_1p_3p_a[1 - p_2p_b - p_4p_b + p_2p_4p_b]$ $C = \frac{p_aq_1q_3}{(q_a + p_aq_1q_3)}$ $D = \frac{p_bq_2q_4}{(q_b + p_bq_2q_4)}$	$\Omega = \frac{[\gamma+A][\gamma+B][\delta+\gamma]}{\gamma^2}$ $p'_1 = \frac{(\gamma)}{(\gamma+AC)}$ $p'_2 = \frac{(\gamma)}{(\gamma+B)}$ $p'_b = \frac{(\gamma+D\delta)}{(\gamma+\delta)}$ $p'_3 = \frac{(\gamma)}{(\gamma+D\delta)}$ $p'_a = \frac{(\gamma+AC)}{(\gamma+A)}$
		$\gamma = p_1p_2p_3p_4p_ap_b[1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4}]$ $B = p_1q_2p_3q_4p_ap_b$ $A = p_2p_3p_b[1 - p_1p_a - p_4p_a + p_1p_4p_a]$ $\delta = p_1p_2p_a[1 - p_3p_b - p_4p_b + p_3p_4p_b]$ $C = \frac{p_aq_1q_4}{(q_a + p_aq_1q_4)}$; $D = \frac{p_bq_3q_4}{(q_b + p_bq_3q_4)}$	$\Omega = \frac{[\gamma+A][\gamma+B][\delta+\gamma]}{\gamma^2}$ $p'_3 = \frac{(\gamma)}{(\gamma+D\delta)}$ $p'_1 = \frac{[\gamma]}{[\gamma+AC]}$ $p'_b = \frac{(\gamma+D\delta)}{(\gamma+\delta)}$ $p'_2 = \frac{(\gamma)}{(\gamma+B)}$ $p'_a = \frac{(\gamma+AC)}{(\gamma+A)}$
		$\gamma = p_1p_2p_3p_4p_5p_ap_b[1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4} + \frac{q_5}{p_5}]$ $B = p_1p_2p_3p_4p_5p_ap_b[\frac{q_3q_4}{p_3p_4} + \frac{q_2q_5}{p_2p_5} + \frac{q_2q_4}{p_2p_4} + \frac{q_1q_5}{p_1p_5}]$ $A = p_2p_3p_5p_b[1 - p_1p_a - p_4p_a + p_1p_4p_a]$ $\delta = p_1p_2p_4p_a[1 - p_3p_b - p_5p_b + p_3p_5p_b]$ $C = \frac{p_aq_1q_4}{(q_a + p_aq_1q_4)}$ $D = \frac{p_bq_3q_5}{(q_b + p_bq_3q_5)}$	$\Omega = \frac{(\gamma+A)(\gamma+B)(\delta+\gamma)}{\gamma^2}$ $p'_1 = \frac{(\gamma)}{(\gamma+AC)}$ $p'_2 = \frac{(\gamma)}{(\gamma+B)}$ $p'_3 = \frac{(\gamma)}{(\gamma+D\delta)}$ $p'_a = \frac{(\gamma+AC)}{(\gamma+A)}$ $p'_b = \frac{(\gamma+D\delta)}{(\gamma+\delta)}$
		$\gamma = p_1p_2p_3p_4p_5p_6p_ap_b[1 + \frac{q_1}{p_1} + \frac{q_2}{p_2} + \frac{q_3}{p_3} + \frac{q_4}{p_4} + \frac{q_5}{p_5} + \frac{q_6}{p_6}]$ $B = p_1p_2p_3p_4p_5p_6p_ap_b\ \frac{q_3q_4}{p_3p_4} + \frac{q_2q_5}{p_2p_5} + \frac{q_2q_4}{p_2p_4} + \frac{q_1q_5}{p_1p_5}$ $+ \frac{q_1q_6}{p_1p_6} + \frac{q_3q_5}{p_3p_5} + \frac{q_2q_6}{p_2p_6}]$ $A = p_2p_3p_5p_6p_b[1 - p_1p_a - p_4p_a + p_1p_4p_a]$ $\delta = p_1p_2p_4p_5p_a[1 - p_3p_b - p_6p_b + p_3p_6p_b]$ $C = \frac{p_aq_1q_4}{(q_a + p_aq_1q_4)}$ $D = \frac{p_bq_3q_6}{(q_b + p_bq_3q_6)}$	$\Omega = \frac{[\gamma+A][\gamma+B][\delta+\gamma]}{\gamma}$ $p'_1 = \frac{[\gamma]}{[\gamma+AC]}$ $p'_2 = \frac{(\gamma)}{(\gamma+B)}$ $p'_3 = \frac{(\gamma)}{(\gamma+D\delta)}$ $p'_a = \frac{(\gamma+AC)}{(\gamma+A)}$ $p'_b = \frac{(\gamma+D\delta)}{(\gamma+\delta)}$

7.9 Conclusion

La factorisation et la réduction sont des méthodes efficaces qui permettent l'évaluation de la fiabilité des réseaux indépendamment de leurs tailles et du nombre de nœuds terminaux. Nous avons démontré que même si les réseaux sont imparfaits ils peuvent aussi être décomposés tout autant que ceux qui possèdent des nœuds non-défaillants. L'algorithme développé dans le cadre de ce travail permet de calculer la fiabilité des réseaux après avoir appliqué des réductions simples et des réductions polygone-à chaine. Nous avons montré par des manipulations élémentaires des probabilités conditionnelles sur les nœuds et les arêtes, comment construire une solution dont le temps d'exécution est à moindre coût. L'idée étant de choisir les arêtes dont l'une de leurs extrémités soit non-défaillante et l'ensemble des transformations nous conduit vers une structure simplifiée d'où l'on peut facilement calculer la fiabilité du réseau. Nous pouvons remarquer sans démonstration préalable que l'algorithme de calcul est polynomial et le nombre de réductions est fini. L'algorithme général et les calculs sur des benchmarks non pu être présentés dans ce travail, cela nécessite une programmation plus poussée pour une implémentation efficace. Aussi, comme perspective, nous envisageons d'étendre ce modèle vers d'autres structures plus complexes et d'écrire un programme plus efficace que celui déjà implémenté sous MatLab qui utilise des structures de données simples pour la mémorisation des calculs intermédiaires et des structures en décomposition.

Bibliographie

Abraham, J. A.,1979, An improved algorithm for network reliability, *IEEE Transactions on Reliability,* Vol. 28, pp. 58–61.

Brace, K., Rudell, R.and Bryant, R., Efficient implementation of a BDD package, *Proceedings of the 27th ACM/IEEE Design Automation Conference*, pp. 40-45.

Bryant, R., 1986, Graph based algorithms for boolean function manipulation, *IEEE Transactions on Computers*, Vol. 35, NO 8, pp. 677-691.

Choi., Jun C.H, 1995, Some variants of polygon-to-reductions in evaluating reliability of undirected networks, *Microelectronics and Reliability*, Vol. 1, NO 35, pp. 1-11.

Deo N. and Medidi M., 1992, Parallel algorithms for terminal-pair reliability, *IEEE Transactions on Reliability*, Vol. 41, pp. 201-209.

Dotson, W. P. and Gobien, J. 0.. 1979, A new analysis technique for probabilistic graphs, *IEEE Transactions Circuits and Systems,* Vol. 26, pp. 855-865.

Fratta, L. and Montanari, U. G., 1973, A boolean algebra method for computing the terminal reliability in a communication network, *IEEE Transactions on Circuit Theory*, Vol. 20, pp. 203-211.

Hardy, G., Lucet, C., and Limnios, N., 2007, K-Terminal network reliability measures with binary decision diagrams, *IEEE Transaction on Reliability*, Vol. 56, NO. 3, pp. 506-515.

Heidtmann, K.D., 1989, Smaller sums of disjoint products by subproduct inversion, *IEEE Transactions on Reliability,* Vol. 38, pp. 305-311.

Institute of Electrical and Electronics Engineers, 1990, IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, New York, NY, 1990.

Kuo, S.Y., Yeh, F.M. and Lin, H.Y., 2007, Efficient and exact reliability evaluation for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 56, pp. 288-298.

Lin, H.Y., Kuo, S.Y. and Yeh, F.M., 2003, Minimal cutset enumeration and network reliability evaluation by recursive merge and BDD, *Proceedings of the Eighth IEEE International Symposium on Computers and Communication*, pp. 1530-1346.

Liu, H. H., Yang, W. T. and Liu, C. C., 1993, An improved minimizing algorithm for the summation of disjoint products by Shannon's expansion, *Microelectronic Reliability*, Vol. 33, NO 4, pp. 599-613.

Locks, M. O. and Wilson, J. M., 1992, Note on disjoint products algorithms, *IEEE Transactions on Reliability*, Vol. 41, pp. 81-84.

Misra. K.B., 1970, An algorithm for the reliability evaluation of redundant networks, *IEEE Transactions on Reliability*, Vol. R-9, pp. 146-151.

Moore and Shannon, 1956, Reliable circuits using less reliable relay, J. Franklin Insti. 262, pp. 191-208..

Moskowitz. F, 1958, The analysis of redundancy networks, *AIEE Trans. Commun. Electron.,* Vol 39, pp 627-632.

Murchland. J., 1973, Calculating the probability that a graph is disconnected, Working paper NO. 18.

Nakazawa. H., 1976, Bayesian decomposition method for computing reliability of oriented network, *IEEE Transactions on Reliability*, Vol R-25, pp 77-80.

Page, L.B. and Perry, J.E., 1988, A Practical Implementation of the factoring theorem for network reliability, *IEEE Transactions on Reliability*, Vol. 37, pp 259-267.

Page, L.B. and Perry, J.E., 1989, Reliability of directed network using the factoring theorem, *IEEE Transactions on Reliability*, Vol. 38, pp 556-562.

Rebaiaia M-L, Ait-Kadi D. and Merlano. A., 2009, A Practical Algorithm for network reliability evaluation based on the factoring theorem — A case study of A generic Radio-communication network, *Journal of Quality*, Vol. 16, N0 5, pp 323-335.

Resende, M.G.C., 1986, A program for reliability evaluation of undirected networks via polygon-to-chain reductions, *IEEE Transaction on Reliability*, Vol. R-35, pp 24-29.

Rosenthal. A., 1974, Computing reliability of complex systems, PhD Thesis, University of California, Berkely.

Rudell, R., 1993, Dynamic variable ordering for ordered binary decision diagrams, *Proceedings of the IEEE International Conference on Computer Aided Design*, pp 42-47.

Satyanarayana. A., Chang M., 1983, Network reliability and the factoring theorems, *Networks*, Vol 13, pp 107-120.

Satyanarayana. A., Wood. R.K, 1985, A linear-time algorithm for computing K-terminal reliability in series-parallel networks, *SIAM J. computing*, Vol 14, pp 818-832.

Sheng-DeWang and Cha-HonSun, 1996, Transformations of star-delta and delta-star reliability networks, *IEEE Transactions on Reliability,* Vol. 1, NO 45, pp 120-126.

Simard, C., 1996, *Contribution au problème d'évaluation de la fiabilité des réseaux*, Mémoire de Maîtrise Sciences, Laval University, Québec, Canada. TJ-7.5-UL-1996.

Tarjan, R.E., 1972, Depth first search and linear graph algorithms, *SIAM Journal of Computing*, Vol. 1, pp 146-160.

Theologou, R., and Carlier, J.G., 1991, Factoring & Reduction for networks with imperfect vertices, *IEEE Transactions on Reliability*, Vol. 40, pp 210-217.

Valian, L.G., 1979, The Complexity of enumerating and reliability problems, *SIAM Journal of Computing*, Vol. 8, pp 410-421.

Wood, R. K.. 1986, Factoring algorithms for computing K-terminal network reliability, *IEEE Transactions on Reliability*, Vol. R-35, pp 269-278.

Yan, L., and Taha, H. A., 1994, A recursive approach for enumerating minimal cutsets in a network, *IEEE Transaction on Reliability*, Vol. 43, pp 383-388.

Yoo Y. B. and Narsingh Deo, 1988, A Comparison of algorithms for terminal-Pair reliability, *IEEE Transactions on Reliability,* Vol. 2, NO. 37, pp 210-215.

General Conclusion

The content of this thesis follows from a real problem of an existing radio-communication network called RENIR. This feature was relevant because the mission of the network was to enable various network users to communicate even if failures could have serious consequences. To address this problem, we conducted a detailed analysis of all components of the system and running simulations to identify communicating path's link which allow two or more users to communicate. This analysis allowed us to generate a model that describes the operations of the network and the reliability of each component. Once the reliabilities are known, we used classical algorithms for determining minimal paths set (MPS) and minimal cuts set (MCS). Despite that these methods are conceptually simple to understand, they are very difficult to exploit, especially for a large network. Therefore we have developped intelligent programs that generate the MPS and MCS for the evaluation of network reliability using binary decision diagrams (BDD). The performance of these programs has been evaluated from a series of tests on networks published in the literature. We were able to find the same results as those published in such papers with less computing time being with a smaller number of iterations.

This contribution allows us to say that the programs developed in this thesis could be used to estimate the reliability of more complex networks. New programs were also developed to assess the reliability of networks using the factoring methods. Again, the developed programs are a significant contribution in the field of evaluation and optimization of network reliability. Having access to these computational tools, we have addressed the problem of optimizing the network reliability. Two important issues were addressed: Maximizing reliability without any constraints and that should be optimized in a specific area. This is the case when we used an approach based on the performance index of Birnbaum, which acts on the most important component for deriving the number of redundant components. On the other hand other indexes have been used to tackle the optimization problem under constraints.

In this thesis, we considered only the case where the components are not repairable and that can have one of two states (up, down). For the network RENIR, it had also to consider the repair of components and the assessment of network availability. It had also to put in place strategies for preventive inspection and replacement of certain components of the network. Note that access to facilities is still not possible because of cold weather and terrain topology. It should be noted that the problem of communication trafic was not considered because data are not available. In addition we have assumed that the software that manages the trafic was cent percent reliable. Also, we note that the problem of common cause failures by far has not been addressed in this thesis, it will be the subject of future works. Several studies are currently underway to address the issues above. Also, other research directions would be developed by considering the following points :

- The stochastic dependency of hardware and software failure that causes links and nodes failure.
- Availability, maintenance and inspection of repairable systems.
- Create database of network structures to render the reliability evaluation process more automatic when dealing with factoring methods.
- Create database for binary decision diagram models to integrating the reliability evaluation and to facilitate its optimization.
- Integrating more techniques for optimizing network reliability as those of Birnbaum's indices by considering constraints domains.
- Extending the software tool for possible sales for interested users.