Exact Methods to compute Network Reliability

CORINNE LUCET (*) JEAN-FRANCOIS MANOUVRIER (*)

Keywords *: reliability , network , enumeration , reduction , factoring , decomposition , treewidth.*

abstract

In this paper, we present and compare some exact methods to resolve the network reliability problems. These problems concern all kinds of networks, such as computer, communication or power networks. When components of the network are subject to random failures, the network may or may not continue functioning after the failures of some components. The probability that the network will function is its reliability. Networks are modeled by a graph G = (V,E) composed of elements that fail independently of each other with known probabilities. The K-terminal reliability problem has been studied extensively. It consists in evaluating the probability that a given subset of vertices, denoted K, is connected. This problem is NP-hard. We propose here to expose the main methods, developed since the 1970's. We first consider the enumeration methods using elementary states, paths or cuts. Then we explain the factoring method performed by the reductions. These allow to treat series-parallel graph in linear time. At last, we present the decomposition method implemented as a table-based reduction algorithm and that allows us to resolve the reliability problem in linear time for graphs with bounded treewidth.

1. Introduction

There are a lot of works in literature devoted to the reliability problem [1][2][3]. For general networks, the K-terminal reliability problem is NP-hard [4]. The network reliability has been approached and resolved with different methods in the literature. The methods that provide an exact reliability are called exact methods, in opposition to the simulation methods that provide an approximate result. The aim of this paper is to present and compare some exact methods to resolve the network reliability problems.

This paper is organized as follows. First, in section 2, we describe the problems and give some definitions. Then in section 3, we present the enumeration methods which are state enumeration and path or cut enumeration. Next in section 4, we explain the factoring method performed by the reduction method, which is linear for series-parallel graphs. Then in section 5, we present the decomposition method, that allows to compute reliability in linear time for graphs with bounded treewidth. Finally, in section 6, we compare the efficiency of these methods.

2. Definitions and notation

Notation

G = (V,E)	a graph with V, the entire set of vertices in G, and E⊂VxV, the entire set of arcs (or edges) in G.
K	a subset of vertices of G, K⊂V. A vertex that belongs to K is called a K-vertex.
p_e and q_e	p_e : the reliability of the component e (e∈V or e∈E); $q_e = 1 - p_e$.
x_e	a boolean variable for the state of the component e∈G (e functions or not).
R(G)	reliability of G.
G_i	a state of G.
G(G_i)	partial graph of G associated with G_i.
Pr(X)	probability for the boolean expression X to be true.
! , + , .	the boolean operators « not », « or » and « and ».
|K| , |V| , |E|	denote the cardinal of the sets K, V, E.
H , L	subgraphs of G.
F	the vertex boundary set between H and L.

A stochastic graph

Our network model is a graph G=(V,E). This graph is directed if E is a set of pairs, i.e. a set of arcs, or undirected if E is a set of unordered pairs, i.e. a set of edges. The graph is said stochastic because each element (vertices, arcs or edges) can fail, statistically independently, with known probability. The failure probability of the element e is denoted q_e ($q_e \in [0,1]$) and its reliability is $p_e=1-q_e$.

(*) UMR CNRS 6599 Heudiasyc , Université de Technologie de Compiègne, B.P. 20529, 60205 Compiègne cedex France,
Tel : 33 (0)3 44 23 44 23 Fax : 33 (0)3 44 23 44 77 e-mails : Corinne.Lucet@hds.utc.fr , JF.Manouvrier@hds.utc.fr

Subgraph and partial graph
A subgraph of a given graph G=(V,E) is a graph G'=(V',E') such that V'⊂V and E'=(V'xV')∩E. A partial graph of a given graph G=(V,E) is a graph G"=(V,E") such that E"⊂E.

Graph state, associated probability and associated partial graph
Each element of the stochastic graph is subject to failure. As there are two states for an element (the element functions or fails), there are $2^{|V|+|E|}$ possible states for the network. One state G_i of the stochastic graph G=(V,E) is denoted $< x_1,x_2,.....,x_{|V|+|E|} >$ where the boolean x_e stands for the state of element e, i.e. $x_e=0$ when element e fails and $x_e=1$ when e functions. The associated probability of G_i is :

$$\text{Proba}(G_i) = \prod_{e \in G} [\, x_e.p_e + (1-x_e).q_e \,]$$

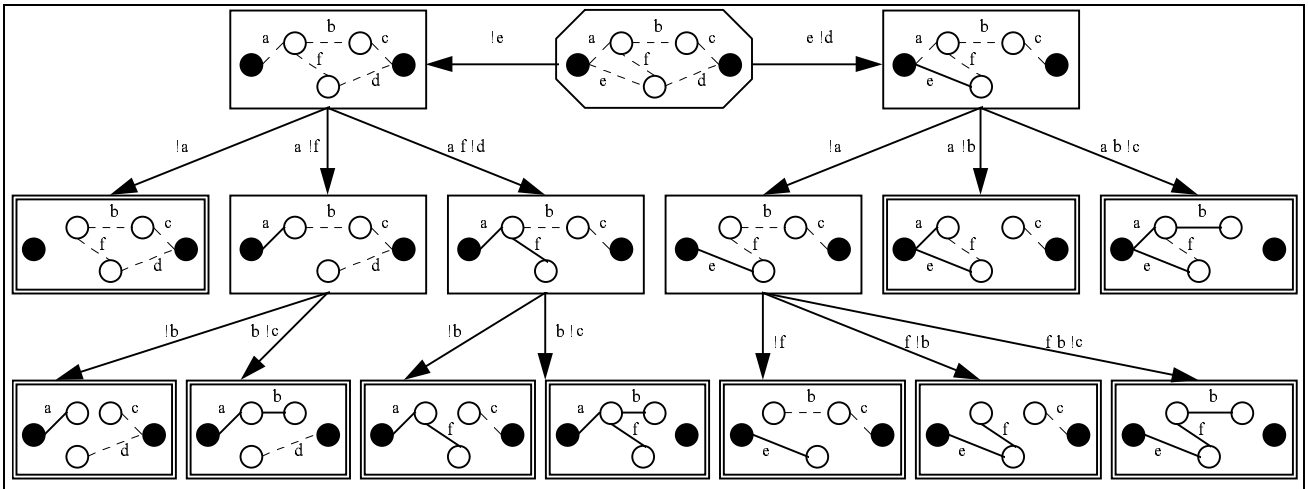
With each state G_i of G=(V,E) is associated a partial graph $G(G_i) = (V",E")$, such that : $u \in V" \Leftrightarrow (u \in V$ and $x_u=1)$ and $(u,v) \in E" \Leftrightarrow ((u,v) \in E, u \in V, v \in V$, and $x_{(u,v)}=x_u=x_v=1)$.
In section 5, we will also consider states and partial graphs of some subgraphs H and L of G, a stochastic graph with perfect vertices. $H_i = < x'_1,x'_2,.....,x'_{|E'|} >$ denotes one state of H=(V',E') and $H(H_i)$ its associated partial graph.

Reliability
The reliability of a network G=(V,E) is the probability that G supports a given operation. We distinguish three kinds of operation and hence, three kinds of reliability : the 2-terminal reliability (also called terminal-pair reliability), the K-terminal reliability and the all terminal reliability. When the operation requires that each pair of vertices are able to communicate via at least one operational path, this is the all terminal reliability. When the operation requires only a few vertices, a subset K of V, to communicate each other, this is the K-terminal reliability. The 2-terminal reliability is the probability that two given vertices, called the source and the sink, can communicate. Note that the 2-terminal and the all terminal reliability are particular cases of the K-terminal reliability. With a directed stochastic graph, the reliability can consider a single source and the other vertices of K are the sinks. In the literature, the usual stochastic graphs are undirected, often with perfect vertices, i.e. each vertex has a reliability that is equal to one.

3. Enumeration

We distinguish among the enumeration methods, the enumeration of states and the enumeration of paths or cuts.

3.1 State Enumeration



R (G) = 1 - ($q_e q_a + q_e p_a q_f q_b + q_e p_a q_f p_b q_c + q_e p_a p_f q_d q_b + q_e p_a p_f q_d p_b q_c + p_e q_d q_a q_f + p_e q_d q_a p_f q_b + p_e q_d q_a p_f p_b q_c + p_e q_d p_a q_b + p_e q_d p_a p_b q_c$)

figure 1 : state factorization with the path method - search for the invalid state sets - 2-terminal reliability

A very basic method to compute the reliability consists in enumerating all the possible states of the stochastic graph and to keep those that provide the functioning of the network or to keep all the failure states, if they are less numerous.

$$\mathbf{R}(G) = \sum_{G(G_i)\ \text{functions}} \text{Proba}(G_i) = 1 - \sum_{G(G_i)\ \text{does not function}} \text{Proba}(G_i)$$

Due to the exponential number of possible states, this formula needs to be improved. Instead of enumerating all the states of the network, a method consists in classifying them in a rooted binary tree. The nodes of this binary tree are partial graphs associated to a set of states, the root stands for the graph with every element that function, and the two branches under a node correspond to a state of a given element : this element functions or not. So the height of this binary tree is |V|+|E| when vertices and edges (or arcs) can fail, and the leaves of the tree are the states of the graph. With this method we look for the failure states of the graph. If a node stands for a failure network, all the states under this node are failure states, and as we know the probability of this set of failure states, we use it to reduce the number of terms in the sum of the previous given reliability formula. Note that a similar method can be used by looking for the operating states; in that case, the root of the binary tree is the network with all element that fail.
So this method looks for the failure states by considering the failure of each element one by one, in an arbitrary order. In order to find these states faster, we can choose an efficient order : for the 2-terminal reliability, the classic technique is to find at each step a path from the source to the sink, and to enumerate the sets of states that prevent this path to function [5][6]. (*FIGURE 1*)

3.2 Path enumeration - Cut enumeration

These methods are composed of two steps. First the enumeration of the paths or the enumeration of the cuts provide a boolean expression, required for the second step, which is the computation of this boolean expression probability. There are two ways to do this second step : the inclusion-exclusion method and the sum of disjoint products technique.

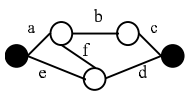
3.2.1 Path enumeration

Another method to compute the network reliability is the enumeration of the minimal paths that provide the network working. The reliability is the probability for the network to have at least a functioning minimal path. These minimal paths are different for each reliability problem. For the 2-terminal reliability, a minimal path is a path from the source to the sink that does not contain another such path. For the K-terminal reliability, the minimal paths are Steiner trees and, for the all terminal reliability, they are spanning trees.

For the 2-terminal reliability, there are several ways to obtain the minimal paths between two vertices [7]. We present here those which use the connection matrix of size |V|x|V| (or adjacency matrix) denoted A with the element $a_{ij} = x_{ij}$ if $(i,j) \in E$, where x_{ij} is a boolean variable standing for the arc (or edge) (i,j) else $a_{ij} = 0$. The matrix $B=A^k$ have for element b_{ij}. b_{ij} is the sum of all the paths from vertex i to vertex j composed of k arcs (or edges). So we can compute all the matrices A^k with $k \leq |V|$ to obtain all minimal paths [8]. A more efficient method is in [9].

For the K-terminal reliability with |K|>2, we quote [10][11] for the search of Steiner trees.

After the enumeration step, a sum is obtained, that is a boolean expression denoted $\Phi(G)$, which terms are minimal paths, and each term is the product of the elements that composed the path : $\Phi(G) = \sum P_i$ with $P_i = \prod e_k$. An example of $\Phi(G)$ is on *FIGURE 2*.
This boolean expression will be used to compute the reliability in section 3.2.3 and 3.2.4 : $\mathbf{R}(G) = \Pr(\Phi(G))$



The four paths : $\Phi(G) = a\,b\,c + a\,f\,d + e\,d + e\,f\,b\,c$
The six cuts : $\Phi_c(G) = !\Phi(G) = !a\,!e + !a\,!d\,!f + !b\,!e\,!f + !b\,!d + !c\,!d + !c\,!e\,!f$

figure 2 : Path (and cut) enumeration - 2-terminal reliability

3.2.2 Cut enumeration

A minimal cut is a minimal set of edges whose failures provide the network failure. Another boolean expression can be obtained with the enumeration of the minimal cuts. This boolean expression, $\Phi_c(G)$, is the complementation of $\Phi(G)$, i.e. the boolean expression of the minimal paths, in the boolean algebra : $\Phi_c(G) = !\,(\Phi(G))$.
An example is on *FIGURE 2*. This inversion principle is discussed in [12].
Once $\Phi_c(G)$ is known, the reliability is computed with : $\mathbf{R}(G) = 1 - \Pr(\Phi_c(G))$

The cut enumeration is essentially used for the 2-terminal reliability. A modified connection matrix can be used as in section 3.2.1. [13] [14]. Other means of enumerating all minimal cuts are in [15][16][17][18].

3.2.3 inclusion-exclusion formula

A method to transform a boolean expression as $\Phi(G)$ or $\Phi_c(G)$ into a probability expression, is to use the Poincare's theorem, also called the inclusion-exclusion method [19][20][1]. Let us consider an example with two alternative minimal paths : $\Phi(G) = P_1 + P_2$ so $Pr(\Phi(G)) = Pr(P_1 \text{ or } P_2) = Pr(P_1) + Pr(P_2) - Pr(P_1 \text{ and } P_2)$. The inclusion-exclusion is a generalization of this principle. With three terms, $\Phi(G) = P_1 + P_2 + P_3$, the formula becomes : $Pr(\Phi(G)) = Pr(P_1)+Pr(P_2)+Pr(P_3) - (Pr(P_1 . P_2) + Pr(P_1 . P_3) + Pr(P_2 . P_3)) + Pr(P_1 . P_2 . P_3)$

Here is the Poincare's formula for M minimal paths:

$$Pr(\Phi(G)) = \sum_{i=1}^{M} Pr(P_i) - \sum_{i_1=1}^{M-1} \sum_{i_2=i_1+1}^{M} Pr(P_{i_1} . P_{i_2}) + \sum_{i_1=1}^{M-2} \sum_{i_2=i_1+1}^{M-1} \sum_{i_3=i_2+1}^{M} Pr(P_{i_1} . P_{i_2} . P_{i_3}) - \dots$$

$$\dots\dots + (-1)^M \sum_{i_1=1}^{2} \sum_{i_2=i_1+1}^{3} \dots \sum_{i_{M-2}=i_{M-3}+1}^{M-1} \sum_{i_{M-1}=i_{M-2}+1}^{M} Pr(P_{i_1} . P_{i_2} . P_{i_3} \dots P_{i_{M-2}} . P_{i_{M-1}}) + (-1)^{M+1} Pr(P_1.P_2.P_3.\dots.P_M)$$

This formula is less efficient than the following one in section 3.2.4, and it is used essentially for an approximation of the reliability by keeping only the first terms of the sum.

3.2.4 sum of disjoint products

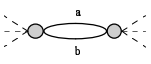
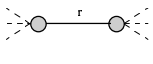
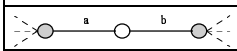
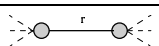
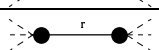
Another method to evaluate $Pr(\Phi(G))$ (or $Pr(\Phi_c(G))$) consists in developing the boolean expression so that each term is an event that does not include another event of the sum, i.e. all the terms are disjoint. This is done by using the following boolean formula : $\Phi(G) = P_1 + P_2 + P_3 + \dots + P_{m-1} + P_m = P_1 + !P_1 . P_2 + !P_1 . !P_2 . P_3 + \dots + !P_1 . !P_2 . !P_3 . \dots . !P_{m-1} . P_m$. Once all the terms of the boolean sum are disjoint, the probability can be easily deduced :
$Pr(\Phi(G)) = Pr(P_1) + Pr(!P_1 . P_2) + Pr(!P_1 . !P_2 . P_3) + \dots + Pr(!P_1 . !P_2 . !P_3 \dots !P_{m-1} . P_m)$
The disjunction must be applied for the elementary variable of the sum. For the development of $\Phi(G)$, the following formula is required : if $P_i = x_{i1} . x_{i2} . x_{i3} . \dots . x_{ip-1} . x_{ip}$
then $!P_i = !x_{i1} + x_{i1} . !x_{i2} + x_{i1} . x_{i2} . !x_{i3} + \dots + x_{i1} . x_{i2} . x_{i3} . \dots . !x_{ip-1} + x_{i1} . x_{i2} . x_{i3} . \dots . x_{ip-1} . !x_{ip}$

Different ways to compute this sum of disjoint products permit to simplify the calculation [21][22][23][2]. The most popular one is the Abraham method.

4. Reduction with factoring

TABLE I : The three series-parallel reductions

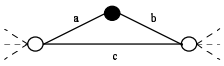
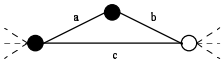
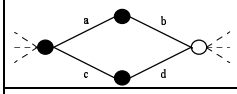
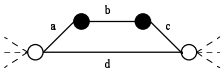
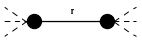
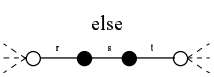
● = a K-vertex ◎ = any vertex ○ = a vertex not in K

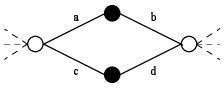
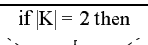
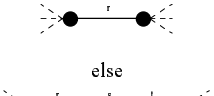
The old part of the network	the new elements provided by the reduction	reduction formulas formula for the new probabilities and reliability	reduction type
		$p_r = 1-q_a q_b$ $\Omega = 1$	parallel reduction
		$p_r = p_a p_b$ $\Omega = 1$	series reduction
		$p_r = p_a p_b / (1- q_a q_b)$ $\Omega = 1 - q_a q_b$ $\lvert K\rvert \leftarrow \lvert K\rvert -1$	series reduction

A reduction is a topological method that can be applied to stochastic undirected graphs [3][24][25]. We first suppose that the vertices are perfect. Indeed, we will consider the graphs with imperfect vertices later. The reduction principle consists in reducing the size of the network G to get a new graph G', such that **R** (G) = Ω . **R** (G'). For this, a part of the network, with a specific topology, is substituted for another one with new edges which failure probabilities depend on the replaced edge probabilities. The reductions have to be applied until the obtained graph G' cannot be reduced. There are three kinds of reductions, the series-parallel reductions (*Table I*), the polygon-to-chain reductions [26] (*Table II*) and the delta-star reductions [27] (star-delta reductions are the reverse reductions) (*Table III*). These

reduction methods are very effective for certain classes of graphs, like the series-parallel graphs, which are reduced to a tree with the series-parallel reductions. In so doing, the reliability can be computed in linear time.

TABLE II : The seven polygon-to-chain reductions

The old part of the network	the substituted elements provided by reduction	reduction formulas	formula for the new probabilities and reliability
		$\alpha = q_a p_b q_c \quad , \quad \beta = p_a q_b q_c$ $\gamma = p_a p_b p_c \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c}\right)$	$p_r = \frac{\delta}{\alpha + \delta}$ $p_s = \frac{\delta}{\beta + \delta}$ $\Omega = \frac{(\alpha + \delta)(\beta + \delta)}{\delta}$
		$\alpha = q_a p_b q_c \quad , \quad \beta = p_a q_b q_c$ $\gamma = p_a p_b p_c \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c}\right)$	
		$\alpha = p_a q_b q_c p_d + q_a p_b p_c q_d + q_a p_b q_c p_d \quad , \quad \beta = p_a q_b p_c q_d$ $\gamma = p_a p_b p_c p_d \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c} + \frac{q_d}{p_d}\right) \quad , \quad \|K\| \leftarrow \|K\| - 1$	
	if |K| = 2 then  else 	$\alpha = q_a p_b p_c q_d \quad , \quad \beta = p_a q_b p_c q_d \quad , \quad \delta = p_a p_b q_c q_d$ $\gamma = p_a p_b p_c p_d \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c} + \frac{q_d}{p_d}\right)$	if |K|=2 then $\Omega = p_b + p_a q_b p_c$ $p_r = (p_b + p_a q_b p_c p_d) / \Omega$ else $p_r = \gamma / (\alpha + \gamma)$ $p_s = \gamma / (\beta + \gamma)$, $p_t = \gamma / (\delta + \gamma)$ $\Omega = (\alpha + \gamma)(\beta + \gamma)(\delta + \gamma) / \gamma$

	if |K| = 2 then  else 	$\alpha = q_a p_b q_c p_d$
		$\beta = p_a q_b q_c p_d + q_a p_b p_c q_d$
		$\delta = p_a q_b p_c q_d$
		$\gamma = p_a p_b p_c p_d \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c} + \frac{q_d}{p_d}\right)$
			if |K| =2 then
			$\Omega = \alpha + \beta + \delta + \gamma$
			$p_r = (\alpha + \delta + \gamma) / \Omega$
			else
			$p_r = \frac{\gamma}{\alpha + \gamma}$,
			$p_s = \frac{\gamma}{\beta + \gamma}$
			$p_t = \frac{\gamma}{\delta + \gamma}$
			$\Omega = \frac{(\alpha + \gamma)(\beta + \gamma)(\delta + \gamma)}{\gamma^2}$

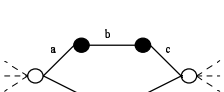
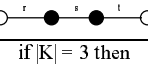
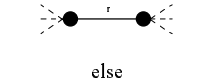
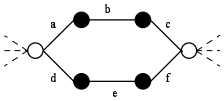
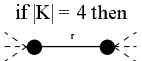
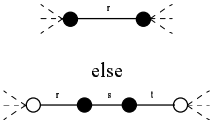
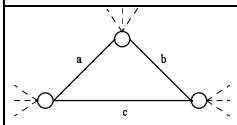
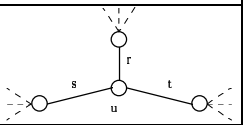
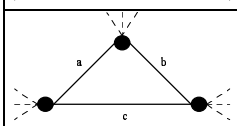
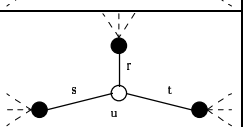
	if |K| = 3 then  else 	$\alpha = q_a p_b p_c q_d q_e$ $\beta = p_a q_b p_c (p_d q_e + q_d p_e) + p_b (q_a p_c p_d q_e + p_a q_c q_d p_e)$ $\delta = p_a p_b q_c p_d q_e$ $\gamma = p_a p_b p_c p_d p_e \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c} + \frac{q_d}{p_d} + \frac{q_e}{p_e}\right)$ $\|K\| \leftarrow \|K\| - 1$	
	if |K| = 4 then  else 	$\alpha = q_a p_b p_c q_d p_e p_f \quad , \quad \delta = p_a p_b q_c p_d p_e q_f$ $\beta = p_a q_b p_c (p_d q_e p_f + q_d p_e p_f + p_d p_e q_f) + p_a p_b q_c p_f (p_d q_e + q_d p_e) + p_a p_b q_c p_d (p_e q_f + q_e p_f)$ $\gamma = p_a p_b p_c p_d p_e p_f \left(1 + \frac{q_a}{p_a} + \frac{q_b}{p_b} + \frac{q_c}{p_c} + \frac{q_d}{p_d} + \frac{q_e}{p_e} + \frac{q_f}{p_f}\right)$ $\|K\| \leftarrow \|K\| - 2$	

TABLE III : delta-star reductions

The old part of the network	the new elements provided by the reduction	reduction formulas formula for the new probabilities and reliability
		$\alpha = p_a p_b + p_a p_c + p_b p_c - 2\, p_a p_b p_c$ $\beta_1 = q_a q_b p_c \quad , \quad \beta_2 = q_a p_b q_c \quad , \quad \beta_3 = p_a q_b q_c$ $p_r = \frac{\alpha}{\alpha + \beta_1} \quad , \quad p_s = \frac{\alpha}{\alpha + \beta_2} \quad , \quad p_t = \frac{\alpha}{\alpha + \beta_3} \quad , \quad p_u = \frac{(\alpha + \beta_1)(\alpha + \beta_2)(\alpha + \beta_3)}{\alpha^2}$ $\Omega = 1$
		

As all graphs cannot generally be totally reduced by reductions, the reduction method is often combined with the factoring method [3][24][25]. The factoring formula divides the reliability problem into two sub-problems :

$$\mathbf{R}(G) = p_e \,.\, \mathbf{R}(G \,/\, \text{edge e functions}) + (1 - p_e) \,.\, \mathbf{R}(G \,/\, \text{edge e fails})$$

The factoring formula must be applied if and only if no reduction on the graph is possible, then, some reductions could be applied in the induced sub-problems.

Let us now consider the stochastic graphs, with imperfect vertices. All reductions can be adapted in case of imperfect vertices [28]. The factoring formula is still valid if we consider that e is composed of an edge and of its two incident vertices, but the failure probabilities of these vertices are modified for the two graphs of the sub-problems [29].

5. Decomposition

The decomposition method permits to solve the reliability problem for some classes of stochastic undirected graph in linear time [30]. At first we will consider the all terminal reliability problem with perfect vertices in section 5.1 and 5.2. The other problems will be studied in section 5.3.

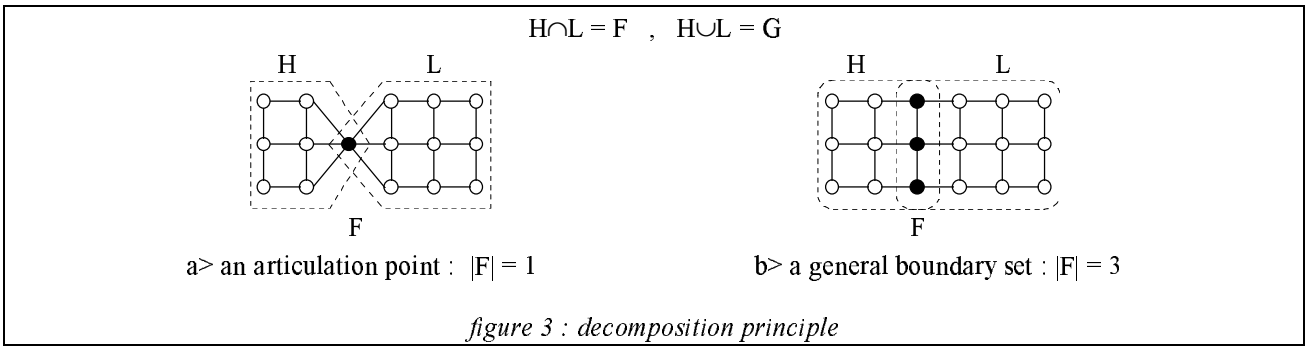


figure 3 : decomposition principle

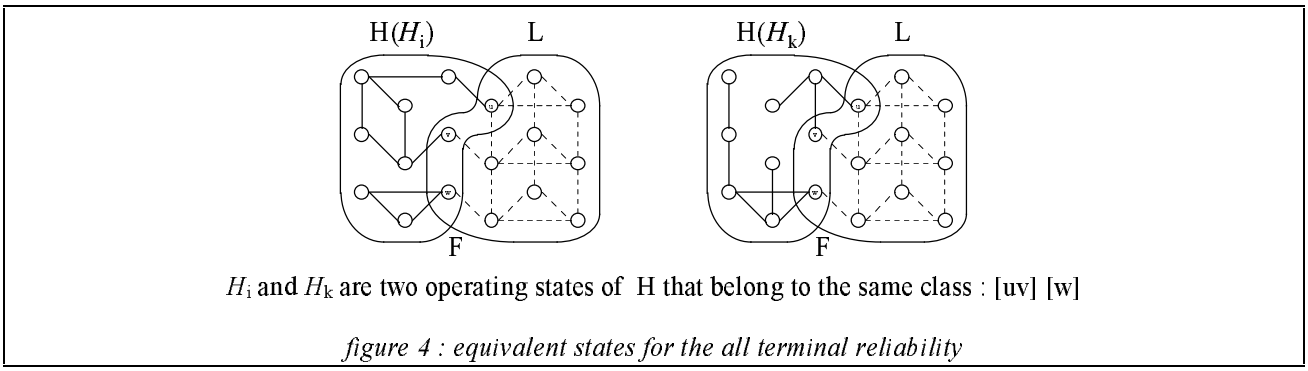


figure 4 : equivalent states for the all terminal reliability

5.1. The principle

The decomposition principle for reliability problems was introduced by Rosenthal in 1977 [31]. It generalizes the basic decomposition principle of a graph G into two subgraphs H and L separated by an articulation vertex (*FIGURE 3.a*), which allows to compute the all terminal reliability with the formula : **R** (G) = **R** (H) . **R** (L)
The decomposition method consists in considering two subgraphs H and L, separated by F, the vertex boundary set, that is a minimal separator, i.e. its removal leaves G disconnected (*FIGURE 3.b*). Each subgraph, H and L, have their set of states, respectively $\{H_i\}$ and $\{L_j\}$, so that the reliability can be written as formula :

$$\mathbf{R}(G) = \sum_{H_i, L_j \ / \ H(H_i) \cup L(L_j) \text{ is connected}} \text{Proba}(H_i) \cdot \text{Proba}(L_j)$$

To use this formula, it is necessary to combine two by two all the states of H and L. To reduce the corresponding number of combinations, first we can eliminate the failure states H_i, i.e. the states such that there exists a vertex in $H(H_i)$ disconnected from the boundary set F, so G cannot be connected. Secondly, we remark that the functioning of G depends only on the connection way of the boundary vertices, via H and via L, so we gather together all equivalent operating states in a same class : two states are equivalent if they provide the same boundary vertex connection

(FIGURE 4). We denote $C_{H,k}$ the k^{th} class of H. These classes are the partitions of the boundary set F, where blocks stand for the connected components. For instance, the different classes of H for a boundary set of three vertices u, v and w are the following ones of H(H_i):

- $C_{H,1}$ = [u v w] : u, v and w are connected via H.
- $C_{H,2}$ = [u v] [w] : u and v are both connected via H, and w is disconnected.
- $C_{H,3}$ = [u w] [v] : u and w are both connected via H, and v is disconnected.
- $C_{H,4}$ = [u] [v w] : v and w are both connected via H, and u is disconnected.
- $C_{H,5}$ = [u] [v] [w] : u, v and w are disconnected via H.

$$\text{Proba}(C_{H,k}) = \sum_{H_i \ / \ H_i \text{ belongs to } C_{H,k}} \text{Proba}\ (H_i)$$

As each class has an associated probability, the previous reliability formula is factored :

$$\mathbf{R}\ (G) = \sum_{C_{H,x},\ C_{L,y} \ / \ C_{H,x} \text{ and } C_{L,y} \text{ are compatible}} \text{Proba}\ (C_{H,x}) \ .\ \text{Proba}\ (C_{L,y})$$

Two classes $C_{H,x}$ and $C_{L,y}$ are compatible if the connectivity of the boundary set given by $C_{H,x}$ and the connectivity of the boundary set given by $C_{L,y}$ provides the connectivity of the whole graph G. The number of classes is function of |F|, so is the efficiency of this method.

5.2. Algorithm implementation

We have seen above the principle of the decomposition which consists in considering two subgraphs H and L of G, and in combining their classes to compute the reliability. Now we see the implementation of this principle with an algorithm [32], that belongs to the table-based reduction algorithm family [33]. At each step of the algorithm, we consider a resolved subgraph H which classes are known and associated probabilities stored in a table, and we enlarge this resolved subgraph by vertex insertions until we have resolved the whole graph. That means that we remove a vertex from L (initially L=G) and add it to H (initially H=∅), using a linear ordering for the vertices. That provides a new boundary set F with its classes and associated probabilities. The table of one step is computed with the table of the previous one. The algorithm is efficient only if the sizes of the boundary sets met during the algorithm are not too large. The larger size met is called the vertex separation number of the linear ordering that is equal to the pathwidth of the corresponding path decomposition [34]. For more efficiency, a similar algorithm could be implemented using a tree decomposition instead of a path decomposition.

So we are confronted with the problem of finding an optimum linear ordering. These problems, the pathwidth and treewidth problems (given a graph, find a tree decomposition or a path decomposition with the smallest width) are NP-hard [35]. Nevertheless, linear solutions exist for graphs with bounded treewidth [36].

The decomposition algorithm can solve the network reliability problem, which is NP-hard, in linear time for a graph with a bounded vertex separation number. It has been proved that some classes of graph problems can be solved in polynomial (or linear time) with such dynamic algorithms using a tree decomposition or a path decomposition with a bounded width [37]. The principle of these algorithms is to use the graph tree or path topology in order to expand a resolved subgraph, until the whole graph is resolved, and to store in memory all partial solutions standing for the resolved elements, i.e. the information required to compute the final solution.

5.3. Adaptation for other reliability problems [32]

This decomposition method can easily be adapted for the general K-terminal reliability problem by considering appropriate classes : These classes are always partition of F, but a boolean for their blocks stand for the presence of a K-vertex in the connected component. Example for F={u,v}, the classes are :

- [u v] , $[u\ v]^k$, [u] [v] , $[u]^k$ [v] , [u] $[v]^k$, $[u]^k$ $[v]^k$ (k means that the block contains some K-vertices)

If we consider the reliability with imperfect vertices, then the classes are modified so that a boundary vertex can fail. Example for F={u,v}, the classes are, for the K-terminal reliability with imperfect vertices :

- [u v] , $[u\ v]^k$, [u] [v] , $[u]^k$ [v] , [u] $[v]^k$, $[u]^k$ $[v]^k$, [u] , $[u]^k$, [v] , $[v]^k$, []

6. Conclusion

The complexity of the enumeration methods is O(|V| . $2^{|V|+|E|}$). Such method can only treat small size networks. The path or cut enumeration are more commonly employed for the 2-terminal reliability. But these enumeration methods cannot be efficient for large networks, i.e. graphs with more than 20 vertices.
The complexity of the reduction method with factoring, using series-parallel reductions for graph with perfect vertices, is : O(|V| !) [3]. Satyanarayana and Wood have shown that the factoring algorithm using reductions is more effective than the classical path or cut enumeration methods in [24] and [25]. This is confirmed by the experimental works of Theologou and Carlier [3]. But, its running time remains prohibitive for large networks, i.e. the treated network did not have more than 30 nodes.
The decomposition method allows to compute the reliability for a bounded pathwidth graph in linear time with the complexity O(|V| . C(Fmax)), where C(Fmax) is a constant that grows exponentially with the pathwidth, denoted Fmax. Carlier and Lucet have worked out and tested this method for the K-terminal reliability problem with imperfect edges and vertices [32]. Their results showed that it is more efficient than factoring using reductions and that it allows to treat real size networks, i.e. the single constraint to apply the decomposition method is the pathwidth, and not the graph size. For example, it computes the reliability of graphs with 140 nodes and a pathwidth of 6 in 6 seconds.

7. References

[1] M. O. Locks, Recent developments in computing of system-reliability, *IEEE Trans. Reliability*, **R-34**, N°5, Dec 1985, 425-435.
[2] S. Rai, M. Veeraraghavan, K. S. Trivedi, A survey of efficient reliability computation using disjoint products approach, *Networks*, **25**, 1995, 147-163.
[3] O. Theologou, Contribution a l'évaluation de la fiabilité des réseaux, Ph.D. Thesis, Université de Technologie de Compiègne, 1990.
[4] M. O. Ball, Complexity of network reliability computations, *Networks*, **10**, 1980, 153-165.
[5] W. P. Dotson, J. O. Gobien, A new analysis technique for probabilistic graphs, *IEEE Trans. Circuits and systems*, **CAS-26**, N°10, Oct 1979, 855-865.
[6] S. Hasanuddin Ahmad, A. T. M. Jamil, A modified technique for computing network reliability, *IEEE Trans. Reliability*, **R-36**, N°5, Dec 1987, 554-556.
[7] Y. Shen, A new simple algorithm for enumerating all minimal paths and cuts of a graph, *Microelectron. Reliab.*, **35**, N°6, 1995, 973-976.
[8] K. B. Misra, T. S. M. Rao, Reliability analysis of redundant networks using flow graphs, *IEEE Trans. Reliability*, **R-19**, N°1, Feb 1970, 19-24.
[9] S. Rai, K. K. Aggarwal, An efficient method for reliability evaluation of a general network, *IEEE Trans. Reliability*, **R-27**, N°3, Aug 1978, 206-211.
[10] S. P. Jain, K. Gopal, An efficient algorithm for computing global reliability of a network, *IEEE Trans. Reliability*, **R-37**, N°5, Dec 1988, 487-492.
[11] M. Piekarski, Listing of all possible trees of a linear graph, *IEEE Trans. Circuit Theory*, **CT-12**, Mar 1965, 124-125.
[12] M.O. Locks, Inverting and minimalizing path sets and cut sets, *IEEE Trans. Reliability*, **R-27**, N°2, Jun 1978, 107-109.
[13] A. Martelli, A gaussian elimination algorithm for the enumeration of cut sets in a graph, *J. Assoc. Comp. Machinery*, **23**, N°1, Jan 1976, 58-73.
[14] D. R. Shier, D. E. Whited, Iterative algorithms for generating minimal cutsets in directed graphs, *Networks*, **16**, 1986, 133-147.
[15] S. Tsukiyama, I. Shirakawa, H. Ozaki, An algorithm to enunerate all cutsets of a graph in linear time per cutset, *J. Assoc. Comp. Machinery*, **27**, N°4, Oct 1980, 619-632.
[16] G. B. Jasmon O. S. Kai, A new technique in minimal path and cutset evaluation, *IEEE Trans. Reliability*, **R-34**, N°2, Jun 1985, 136-143.
[17] S. Hasanuddin Ahmad, Simple enumeration of minimal cutsets of acyclic directed graph, *IEEE Trans. Reliability*, **R-37**, N°5, Dec 1988, 484-487.
[18] L. Yan, H. A. Taha, T. L. Landers, A recursive approach for enumerating minimal cutsets in a network, *IEEE Trans. Reliability*, **R-43**, N°3, Sep 1994, 383-388.
[19] J. A. Buzacott, J. S. K. Chang, Cut-set intersections and node partitions, *IEEE Trans. Reliability*, **R-33**, N°5, Dec 1984, 385-389.
[20] J. A. Buzacott, Node partition formula for directed graph reliability, *Networks*, **17**, 1987, 227-240.
[21] J. A. Abraham, An improved algorithm for network reliability, *IEEE Trans. Reliability*, **R-28**, N°1, Apr 1979, 58-61.
[22] F. Beichelt, L. Spross, An improved Abraham-Method for generating disjoint sums, *IEEE Trans. Reliability*, **R-36**, N°1, Apr 1987, 70-74.
[23] M. O. Locks, A minimizing algorithm for sum of disjoint products, *IEEE Trans. Reliability*, **R-36**, N°4, Oct 1987, 445-453.
[24] A. Satyanarayana and M. Chang, Network reliability and the factoring theorem, *Networks*, **13**, 1983, 107-120.
[25] R. K. Wood, A factoring algorithm using polygon-to-chain reductions for computing K-terminal network reliability, *Networks*, **15**, 1985, 173-190.
[26] M. S. Choi, C. H. Jun, Some variant of polygon-to-chain reductions in evaluating reliability of undirected network, *Microelectron. Reliab.*, **35**, N°1, 1995, 1-11.
[27] J. P. Gadani, System effectiveness evaluation using star and delta transformations, *IEEE Trans. Reliability*, **R-30**, N°1, Apr 1981, 43-47.
[28] M. K. F. Lai, Polygon-to-chain reductions work for networks with imperfect vertices, *Microelectron. Reliab.*, **34**, N°2, 1994, 267-274.
[29] O. Theologou and J. Carlier, Factoring and reductions for networks with imperfect vertices, *IEEE Trans. Reliability*, **R-40**, N°2, 1991, 210-217.
[30] J. Carlier and C. Lucet, A decomposition algorithm for network reliability evaluation, *Discrete Applied Mathematics*, **65**, 1996, 141-156.
[31] A. Rosenthal, Computing the reliability of complex networks, *SIAM J.Appl.Math.*, **32**, N°2, 1977, 384-393.
[32] C. Lucet, Méthode de décomposition pour l'évaluation de la fiabilité des réseaux, Ph.D. Thesis, Université de Technologie de Compiègne, 1993.
[33] S. Arnborg, Efficient algorithms for combinatorial problems on graphs with bounded decomposability - A survey, *BIT*, **25**, 1985, 2-23.
[34] N. G. Kinnersley, The vertex separation number of a graph equals its path width, *Inform. Proc. Letters*, **42**, 1992, 345-350.
[35] S. Arnborg, D. G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM J. Alg. Disc. Meth.*, **8**, N°2, 1987, 277-284.
[36] H. L. Bodlaendler, A linear time algorithm for finding tree-decompositions of small treewidth, *SIAM J. COMPUT.*, **25**, N°6, 1996, 1305-1317.
[37] H. L. Bodlaendler, Dynamic programming on graphs with bounded treewidth, *proc.15th ICALP'88*, **317**, 1988, Springer Verlag, 105-118.