

Optimisation du diagnostic des pannes des systèmes industriels complexes : Approche par génération de Coupes Minimales basée sur les diagrammes de décision binaire

Mohamed-Larbi Rebaiaia
*CIRRELT, département de génie
mécanique, Université Laval,
Québec (QC), Canada*

Mohamed-larbi.rebaiaia@gmc.ulaval.ca

Daoud Ait-Kadi
*CIRRELT, département de génie
mécanique, Université Laval,
Québec (QC), Canada*

Daoud.aitkadi@gmc.ulaval.ca

RÉSUMÉ — Le problème de la détermination des séquences de tests pour détecter les défauts et les possibles pannes dans l'ingénierie des systèmes est un problème majeur. En effet, chaque composant est assujéti à des défaillances aléatoires. Cependant, la vérification de l'état d'un composant ou d'un sous-système est effectuée à l'aide de tests. Cependant, toute solution qui permet d'obtenir un pronostic fiable et économique est essentielle, cela permet à toute industrie d'être viable, sécuritaire et compétitive. Dans le cadre de cette étude, nous proposons un modèle de calcul qui permet de nous fournir des séquences de tests simples à générer et optimales sur la base du coût de déploiement. Cette méthode de calcul est articulée sur un algorithme efficace qu'on propose pour déterminer les coupes minimales. L'algorithme procède en déterminant l'ensemble des chemins minimaux (MPS) et à partir desquels des coupes minimales (MCS) sont générées par l'utilisation du concept de diagrammes de décision binaire réduits et ordonnés et de manipulations de termes. Le processus de manipulation consiste en une série de transformations, de réductions et d'opérations de filtrage des coupes minimales ainsi générées. L'approche a permis de réduire le temps de calcul et l'espace mémoire et a été appliquée à l'évaluation de la fiabilité de plusieurs réseaux complexes. En outre, nous avons testé la méthode proposée en utilisant un algorithme qui permet de détecter les séquences de tests d'un système industriel bien défini. Les résultats obtenus sont intéressants et ouvrent la voie vers d'autres systèmes à tester plus complexes.

MOTS-CLEFS — Évaluation des performances, ensemble de coupures minimales, fiabilité des réseaux à 2 terminaux.

ABSTRACT — The problem of determining test sequences to detect defects and possible failures in systems engineering is a major problem. Indeed, each component is subject to random failures. However, checking the status of a component or a subsystem is performed using testing. However, any solution that provides a reliable and economical prognosis is essential, this allows any industry to be viable, safe and competitive. As part of this study, we propose a calculation model which allows to provide us with test sequences that are simple to generate and optimal based on the deployment cost. This calculation method is based on an efficient algorithm that we propose to determine the minimal cuts set. The algorithm proceeds by determining the set of minimal paths (MPS) and from which minimum cuts (MCS) are generated using the concept of reduced and ordered binary decision diagrams and term manipulations. The manipulation process consists of a series of transformations, reductions and filtering operations of the generated minimal cuts. The approach reduces the calculation time and memory space and is applied to the reliability assessment of several complex networks. Furthermore, we tested the proposed method using an algorithm that can detect test sequences of a well-defined industrial system. The results obtained are interesting and open the way to other, more complex systems to test.

KEYWORDS — Performances Evaluation, Minimal Path set Minimal Cuts set, Network Reliability Determination.

Nomenclature :

U	Ensemble de nœuds (sommets) d'un graphe G
V	Ensemble de liens (arcs, arêtes) entre deux nœuds d'un graphe G quelconque
G(U, V)	Représentation graphique d'un système dite modèle ou graphe
Arc	Grappe orienté

Arête	Graphe non-orienté
Path	Chemin : Est un ensemble de liens d'un graphe (arcs, arêtes) reliant deux nœuds distincts d'un graphe
MP	Chemin minimal dont les liens et les nœuds ne se répètent pas
Cut	Coupe : Est un ensemble de liens coupant le graphe en deux parties distincts (deux sous-graphes)
MC	Coupe minimale : est une coupe formée d'un nombre minimum de liens pouvant couper le graphe en deux
MPS	Minimal Paths Set (Ensemble des chemins optimaux)
MCS	Minimal Cuts Set (Ensemble des coupes minimales) Minimal Cut Set.
BDT	Binary Decision Tree (Arbre Binaire de Décision)
BDD	Binary Decision Diagram (Diagramme Binaire de Décision)
OBDD	Ordered Binary Decision Diagram (BDD dont les variables de décision sont ordonnées)
ROBDD	Reduced Ordered Binary Decision Diagram (OBDD réduit)
DFS	Depth First Search (Algorithme de parcours en profondeur d'un graphe d'un nœud racine vers d'autres nœuds)
BFS	Breadch First Search (Parcours en largeur d'un graphe d'un nœud racine vers d'autres nœuds)
BDF	Bloc-diagramme- de fiabilité: est un graphe utilisé pour modéliser le fonctionnement d'un système

38

39

I. INTRODUCTION

40

Les systèmes industriels sont en général très complexes et critiques, vu qu'ils sont composés d'un grand nombre de composants qui parfois sont hétérogènes et dépendants. Compte tenu d'un fonctionnement prolongé, les propriétés de certains de leurs composants peuvent se dégrader graduellement jusqu'à un niveau critique, qui dans la plupart des cas, peut entraîner l'apparition d'un ou de plusieurs défauts et même des défaillances incontrôlées. Par conséquent, les exigences du point de vue de fiabilité et de sécurité, des stratégies de surveillance et de contrôle de ces défauts et pannes doivent être mises en place pour assurer le bon fonctionnement du système qui doit se libérer des arrêts incontrôlés. Cependant, lorsqu'un défaut est constaté, un processus d'isolation des défauts doit être utilisé pour trouver l'emplacement et le type de défaut. Dans ce cas de figure, on parle alors de pratiques de diagnostic des défauts et déterminer jusqu'à quel seuil il est question d'accepter le niveau de défaut qu'on appelle communément par tolérance aux pannes. Au cours des dernières décennies, des efforts de recherche considérables ont été déployés pour trouver des approches systématiques du diagnostic des pannes et du contrôle tolérant aux pannes dans les systèmes industriels complexes (Sunan Huang et al. (2020)). Il a été démontré que l'utilisation de modèles de processus adéquats peut permettre un diagnostic automatique précoce des défauts à moindre coût. Contrairement aux méthodes et technique de détection des pannes avant qu'elles arrivent, telle que l'inspection par différents moyens et technologies telles que la mise en place de capteurs de différentes natures, les modèles statistiques de prédiction, les modèles d'intelligence artificielle (Machine-Learning, deep-Learning et réseaux de neurones artificiels, etc. (<https://www.ibm.com/topics/machine-learning> (consulté le 24-06-2024))), la solution que nous proposons est simple est facile à implémenter et à être utilisée en ligne. Il s'agit de générer la séquence de tests qui permet de localiser rapidement le(s) composant(s) responsable(s) de la défaillance du système. Cette méthode suppose que nous avons les outils nécessaires pour déterminer les coupes minimales déterminées à partir du graphe représentant le système industriel à tester. La détermination des coupes minimales n'est pas du tout un exercice facile surtout lorsque le système en question est très complexe en termes de la grande dimension de ses nœuds et liens. C'est dans cette optique que nous proposons un algorithme permettant de générer les coupes minimales d'un réseau après l'utilisation du concept de ROBDD (Bryant (1992), Hardy et al. (2007), Rudell (1993), Kwassau et al. (2020)).

62

Les R-réseaux (R : pour fiabilité) sont un modèle de graphe représentatif des systèmes. Aussi appelés bloc-diagramme de fiabilité (BDF) surtout utilisés en sûreté de fonctionnement pour représenter l'état de marche d'un système en fonction des états de marche de ses composants. Ils sont stochastiques car chaque lien entre deux nœuds prend sa valeur dans le domaine booléen et les événements sont distribués aléatoirement dans $(0,1)^n$. Ils sont des graphes qui peuvent être orientés ou non. Dans ce domaine $(0,1)^n$, « 0 » correspond au mode de défaillance et « 1 » au mode de fonctionnement, et n représente le nombre de nœuds. Les R-réseaux ne sont utilisés que pour déterminer la fiabilité des systèmes. Une abondante littérature relative à l'évaluation de la fiabilité des réseaux a fait l'objet de nombreuses publications et concerne les systèmes physiques tels que les systèmes d'alimentation électrique, les réseaux de télécommunication et les systèmes de circulation et de transport (Rebaiaia et Ait-Kadi, (2009, 2010, 2010 bis, 2011, 2013) ; Hardi *et al.*, 2007, Ali Mohamed Rushdi et al. (2021)).

71

En règle générale, les ingénieurs en fiabilité modélisent la connectivité physique des composants d'un système (exemple : industriel) à l'aide d'un réseau. Mathématiquement, un réseau est un graphe $G(U, E)$ dans lequel les arêtes E représentent les composants (par exemple, les périphériques, les ordinateurs, les routeurs, les machines de fabrication, etc.) et les nœuds U représentent les interconnexions, c'est-à-dire l'ensemble des liens reliant les nœuds.

75

Le problème de l'analyse de la fiabilité des réseaux a fait l'objet de nombreuses productions scientifiques (Rebaiaia, 2011; Rebaiaia et al. (2010), (2015), (2017)), Kuo *et al.*, (2007) ; Al-Ghanim, (1999) ; Rauzy, (1993) ; Yeh, (2007)). Elle consiste à évaluer la fiabilité à 2-terminaux, K -terminaux et tous-terminaux. La lettre K représente le nombre de nœuds de n'importe quel sous-réseau du réseau R. La théorie générale, a longuement discuté deux techniques ; méthodes exactes et approximatives (Locks, 1992). Les méthodes exactes utilisent le concept de MPS/MCS (Jasmon and Kai (1985), Rebaiaia (2011)). La détermination du MCS est essentielle non seulement pour évaluer les indices de fiabilité, mais aussi pour étudier les différents scénarios afin de trouver, par exemple, les composants redondants qui pourraient être ajoutés pour améliorer la fiabilité du

81

point de charge, ou bien la partie faible du réseau susceptible de provoquer la défaillance du réseau. Autrement dit, un MCS est un ensemble de composants d'un système qui, en cas de défaillance, provoque la défaillance du système, mais lorsqu'aucun composant de l'ensemble n'est défaillant, il ne provoque aucune défaillance du système. L'énumération de tous les MCS peut être une solution meilleure en termes de rapidité de calcul par rapport au nombre des MPS si ce dernier est plus grand. Un exemple de ce type de préférences est le réseau 2x100 qui a 2⁹⁹ chemins et seulement 10000 coupes (Kuo *et al.*, 2007). Un autre exemple est le réseau complet avec 10 nœuds qui génère 109601 chemins minimaux et seulement 256 coupes. Dans les algorithmes existants (Jason et Kai, 1985 ; Yan *et al.*, 1994), les chemins minimaux sont déduits du graphe à l'aide d'algorithmes récursifs simples et systématiques qui garantissent que les chemins générés sont minimaux. L'énumération des MCS est plus problématique que celle des MPS car elle nécessite des mathématiques avancées, telles que la théorie des ensembles, l'algèbre de Boole et la manipulation matricielle. Dans l'article de Locks et Wilson (1992), les auteurs ont proposé une méthode simple pour générer des MPS directement à partir de MCS, et vice-versa. Cette méthode commence par l'inversion de l'expression de fiabilité accomplie par une méthode récursive combinant une application en 2 étapes des axiomes de De-Morgan. Yan *et al.*, (1994) ont proposé un algorithme d'étiquetage récursif pour déterminer tous les MCS dans un réseau orienté. Ils ont utilisé une approche adaptée des méthodes de programmation dynamique. L'algorithme produit tous les MCS, puis élimine tout MCS redondant par simple comparaison. Jasmon *et al.* (1985) utilisent un algorithme qui procède en déduisant d'une part, les ensembles de coupures de liens des ensembles de coupures de nœuds, et d'autre part, en déterminant les chemins minimaux de base en utilisant la décomposition du réseau. Dans ce cadre d'étude, Yeh (2007) a présenté un algorithme simple pour trouver tous les MPS entre les nœuds source et récepteur. Il est basé sur la fonction génératrice universelle. Plus récemment, Rebaiaia *et Ait-Kadi* (2012) ont proposé un algorithme élégant et rapide pour dénombrer les MPS à l'aide d'une technique DFS modifiée (DFS : Depth First Search (Algorithme de recherche en profondeur d'abord)). La procédure utilise chaque chemin découvert pour générer de nouveaux MPS à partir de sous-chemins. La procédure ci-dessus est répétée jusqu'à ce que tous les chemins minimaux soient trouvés (MPS). L'algorithme n'a pas du tout produit de MPS redondant.

Cet article présente une technique efficace pour déterminer toutes les MCS d'un réseau de graphes dirigé ou non dirigé donné en utilisant une approche basée sur la représentation de diagrammes de décision binaires (BDD : Binary Decision Diagrams). Le premier intérêt des BDDs est qu'ils codent des termes disjoints. Cela permet de simplifier énormément le calcul probabiliste comme la formule de Poincaré car elle est réduite à son premier terme (pour plus de détails sur la formule de Poincaré, voir Rebaiaia (2011)). C'est pourquoi cette approche est si efficace pour effectuer des calculs probabilistes exacts. L'algorithme procède en deux étapes. La première action consiste à déterminer les MPS à l'aide d'un algorithme de recherche rapide en profondeur (DFS : depth first search). La seconde action consiste à obtenir les MCS en manipulant le BDD ordonné réduit (ROBDD : reduced ordered binary decision diagram) des MPS. Le processus de manipulation consiste en une série de transformations, de réductions et d'actions de filtrage.

Le document est structuré comme suit. La section 2 présente quelques préliminaires connexes. La section 3 détaille le principe des manipulations des BDD. Les sections 4 et 5 donnent respectivement l'algorithme de génération de MCS à partir de ROBDD et son efficacité de calcul illustrée à l'aide de certains réseaux de référence. Dans la section 6, nous détaillerons une méthode mathématique qui permet d'établir le diagnostic des pannes par séquence de tests qui soit optimale en matière de coûts. Cette méthode est tirée des travaux de Gao et Ait-Kadi (1994). Ensuite cette méthode est appliquée à un cas d'étude pour montrer la facilité de son application. Finalement la section 8 conclue cet article.

II. PRÉLIMINAIRES

Prenons l'exemple d'un système industriel composé de m composants numérotés de 1 à m . Chacun de ces composants peut être actif (en cours d'exécution) ou inactif (défaillant). Supposons que la variable x_i représente un composant, et soit X le vecteur d'état. Dans le domaine de l'ingénierie de la fiabilité, un système peut être modélisé par un réseau (graphe) et sa fiabilité est générée à partir de la fonction de structure correspondante qui est identifiée comme la fonction mathématique attendue où les vecteurs sont les MPS ou bien les MCS (voir Rebaiaia, 2011 pour plus de détails). On parlera alors de diagramme de fiabilité qui est une représentation sous forme d'un graphe muni d'une entrée et d'une sortie, où tous les chemins sont formés d'une suite de composants et sont tels que « le système fonctionne si tous les composants d'au moins un chemin de ce graphe fonctionnent » : c'est une condition suffisante pour le fonctionnement d'un système. Un exemple est illustré comme suit :

Considérons le réseau de la Figure 1. (a) à partir duquel les MPS et MCS sont générés et ils sont comme suit :

$$\text{MCS} = \{C_1 = \{x_1, x_2\}; C_2 = \{x_1, x_5\}; C_3 = \{x_2, x_3, x_4\}; C_4 = \{x_4, x_5\}\}$$

$$\text{MPS} = \{P_1 = \{x_1, x_4\}; P_2 = \{x_2, x_5\}; P_3 = \{x_1, x_3, x_5\}\}$$

x_i : représente l'état d'un composant i et P_1 , P_2 et P_3 , et C_1 , C_2 , C_3 et C_4 sont respectivement les MPS et MCS dérivés du réseau de la Figure. 1 (a).

132 La fonction de structure $\Phi(X(t))$ du système est équivalente au graphe série-parallèle (Figure 1 (b)). Elle peut s'écrire
 133 comme suit :

134
$$\Phi(X(t)) = (1 - (1 - x_1)(1 - x_4))(1 - (1 - x_2)(1 - x_5))(1 - (1 - x_1)(1 - x_3)(1 - x_5))$$

135 Après décomposition de la fonction de structure et utilisation des axiomes de simplification de De Morgan, les variables d'état
 136 sont substituées par les composantes fiabilité, la fiabilité du système est calculée comme suit :

137
$$R(t) = \Pr(\Phi(t) = 1) = p_1 p_4 + p_2 p_5 + p_1 p_3 p_5 - p_1 p_2 p_4 p_5 - p_1 p_3 p_4 p_5 - p_1 p_2 p_3 p_5 + p_1 p_2 p_3 p_4 p_5$$

138 Où $\Pr(\cdot)$ est la fonction de probabilité et p_i pour $i = 1, \dots, 5$ sont respectivement la probabilité que la composante i soit UP et
 139 $q_i = 1 - p_i$ DOWN.

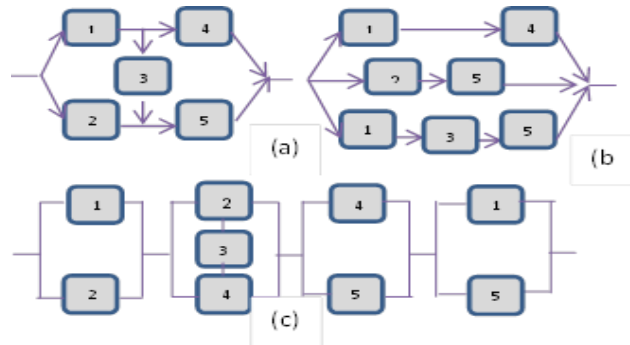


Fig.1. (a) Structure du système, (b) Structure de fiabilité: cas MPS, (c) Structure de fiabilité: cas MCS.

III. APPROCHE PAR DIAGRAMME DE DÉCISION BINAIRE

144 Des techniques expérimentales ont été appliquées pour réduire la fiabilité de l'expression du réseau, mais elles ne sont pas
 145 adéquates pour simplifier l'expression booléenne du réseau. L'une des solutions efficaces a été introduite par Bryant (1992), qui
 146 a été le premier à coder le théorème de Shannon en utilisant une nouvelle formule d'interprétation appelée ITE pour If Then
 147 Else, qui est utilisée pour représenter et manipuler la formule logique sous forme de diagrammes de décision binaires.
 148 L'implémentation et la manipulation des algorithmes BDD est composée de trois procédures, *restrict*, *apply* et *ite*. Le problème
 149 avec la représentation BDD malgré leur efficacité est que leur taille augmente de façon exponentielle en raison d'une mauvaise
 150 déclaration d'ordre entre les variables. Pour surmonter partiellement ce problème, Ruddell (1993) a utilisé un algorithme de
 151 programmation dynamique pour réduire la taille du BDD. Il a également démontré que l'ordre des variables d'un BDD est NP-
 152 Complet. Le principe 'BDD' a été utilisé dans de nombreux domaines de l'ingénierie des systèmes pour simplifier les
 153 expressions booléennes. Rauzy (1993) a été le premier à appliquer les BDD pour évaluer la fiabilité des réseaux.

154 La représentation et la simplification d'une expression booléenne procèdent en 3 étapes comme suit :

- 155 - Construisez l'arbre de décision binaire (BDT) associé à l'expression mathématique de la structure du graphe.
- 156 - Transformez le BDT en BDD en appliquant les règles suivantes :
 - 157 a- Fusionnez les feuilles équivalentes de l'arbre de décision binaire ainsi construit.
 - 158 b- Fusionnez les nœuds isomorphes.
 - 159 c- Élimination des tests redondants
- 160 - Transformez le BDD en BDD ordonné (OBDD) par un choix judicieux sur les variables (voir. Rudell, 1993).

162 L'OBDD peut être réduit à un ROBDD en éliminant à plusieurs reprises toutes les instances de nœuds dupliqués et
 163 redondants à l'aide d'une procédure ascendante de traversée de graphes. Si deux nœuds sont des doublons, l'un d'eux est
 164 supprimé et tous ses pointeurs entrants sont redirigés vers son doublon. Si un nœud est redondant, il est supprimé et tous les
 165 pointeurs entrants sont redirigés vers son seul enfant.

166 Pour pallier cette lacune, Bryant, 1992 a suggéré de représenter la procédure de décomposition de Shannon comme une
 167 fonction ITE (si, sinon), qui à son tour est exprimée par la relation suivante :

168
$$f = ite(x, F_1, F_2) = xF_1 + xF_2$$

169 Deux « pseudo-codes » modifiés ont été écrits pour simuler les fonctions *ite* et *Apply* de Bryant (voir Rebaiaia (2011) et
 170 Rebaiaia et Ait-Kadi (2010)).

IV. GENERATION DE COUPES MINIMALES (MCS) A PARTIR DE ROBDD

Tel que décrit par Locks *et* Wilson (1992), un polynôme inverse du polynôme de chemin peut être obtenu en complétant le polynôme donné et en utilisant les lois de De Morgan. Si l'on transpose l'idée de générer un MCS par inversion telle qu'introduite par Locks *et* Wilson (1992) et Shier *et* Whited (1985), on peut utiliser le modèle généré par la représentation graphique de BDD.

La nouvelle idée est d'abord expliquée à l'aide du réseau de la Figure 1. (a) pour montrer comment on peut générer des coupes minimales en utilisant simplement une procédure de parcours du graphe en profondeur d'abord DFS (Rebaiaia, 2011). Cet algorithme visite les nœuds de graphe BDD représentant les variables de décision booléennes et les arêtes.

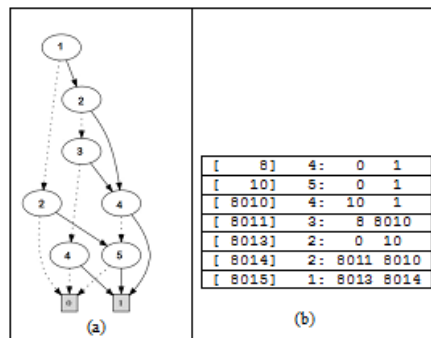
Considérons le réseau de Koninsberg donné à la Figure 1. (a) et son ROBDD correspondant construit à partir du MPS (Figure 2. (a)). La première étape consiste à essayer de trouver la coupe minimale en parcourant l'arborescence du graphe vers le bas. La deuxième étape procède à l'enlèvement des coupes des branches. Dans la troisième phase, l'algorithme supprime toutes les coupes minimales redondantes pour construire le MCS.

La procédure suivante fonctionne pour générer des MCS à l'aide d'un algorithme de recherche en profondeur et d'informations de données tirées de la matrice de la Figure 2. (b).

Procédure *Generation of MCS*

- Placez le nœud carré au-dessus d'une pile 1 */** enregistre les visites DFS aux nœuds ROBDD **/*.
- Placez le nœud carré au-dessus d'une pile 2 */** enregistre les nœuds coupés.
- Placez sur le dessus de la pile 1 tous les nœuds ascendants de la variable supérieure de la pile.
- Placez le nœud en haut de la pile 1 au-dessus de la pile 2, si l'arête (lien) est en pointillés.
- Continuez jusqu'à ce que la variable atteigne le nœud racine.
- Si c'est le cas, une coupure a été trouvée. Écrivez le contenu de la pile 2 sous la forme d'une ligne d'une matrice. Supprimez la variable supérieure de la pile 1 et de la pile 2.
- Continuez la procédure jusqu'à ce que la pile 1 soit vide.
- Appliquez le processus de filtrage en supprimant tous les chemins redondants (coupes) à l'aide de la matrice de chemins (coupes)
- Affichez la matrice MSC.

fin procédure



La fig. 2. (a) : Le réseau BDD de la figure 1. (a) et (b) : son code de représentation en mémoire.

La procédure de filtrage supprime les coupes redondantes de l'ensemble de toutes les coupes. Il procède comme suit :

- 1- Triez la matrice CS dans un ordre croissant en fonction de la taille de chaque vecteur (nombre de variables) ;
- 2- Prenez le premier vecteur et comparez-le avec chacun des autres vecteurs ;
- 3- Si les membres de l'intersection sont égaux au premier vecteur, alors supprimez le vecteur réel de la matrice CS ;
- 4- Itérez en utilisant les autres vecteurs de la matrice.

La pseudo-description suivante montre le traitement de la procédure de filtrage comme expliquée ci-dessous :

Procédure *De_Filtrage (CS, MCS)*

```

n = longueur(CS) ; /* taille du vecteur matriciel */ ; CS : vecteur cut set
m = taille(CS) ; /* taille du vecteur matriciel
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) est un vecteur redondant)
            Supprimez le vecteur CS(j,k) ;
        endif
    end for
end for
MCS = CS ; imprimez le MCS

```

Fin_de_procedure

V. RESULTS EXPERIMENTAUX POUR LA GENERATION DE COUPES MINIMALES

Les algorithmes proposés ont été implémentés dans MatLab et Java Jdk. Une interface communicante a été écrite pour faciliter le transfert des données et des résultats entre le système MatLab et les packages Java fonctionnant sous jGRASP. Ce dernier est une interface graphique écrit en Java JDK. Le système d'exploitation est de 32 bits Windows de Microsoft. Il s'agit d'un ordinateur portable HP équipé d'un processeur Intel(R) core (TM) 2 Duo de 1,67. Les réseaux de référence de la figure 3 ont été utilisés et les résultats sont présentés dans le tableau suivant. Tous les réseaux sont à 2 terminaux (un nœud source et un autre terminal). Ils constituent un benchmark utilisé dans différents articles de publication. Nous pouvons remarquer à partir de ces résultats, que la valeur du temps d'exécution est intéressante malgré le fait que les performances des caractéristiques de la machine ne sont pas élevées. L'importance de ce travail montre l'efficacité des algorithmes développés dans cet article. Il est certain que si le processeur était plus puissant et que l'espace mémoire était plus large, le programme pourrait facilement gérer des réseaux plus complexes.

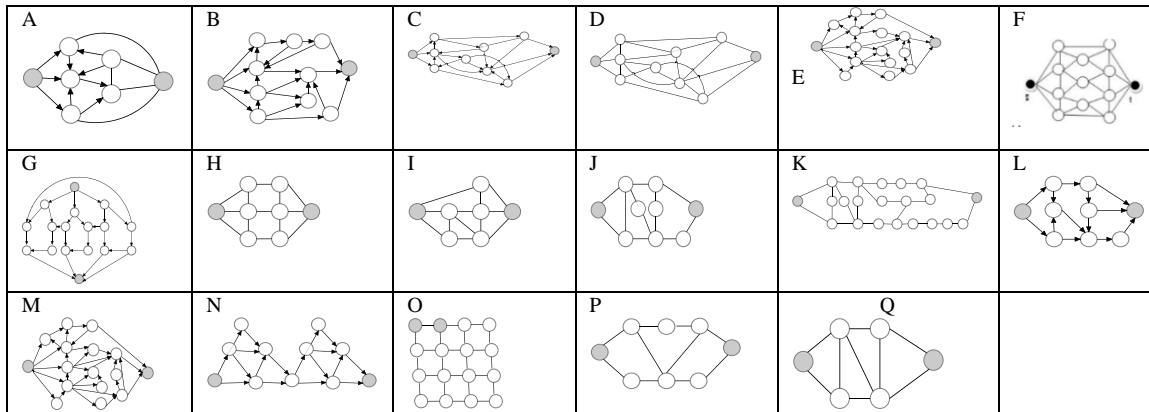


Fig.3. Réseaux Tests de référence

Table 1. Nombre de coupes minimales et le temps d'exécution relatifs à chaque réseau

Nets	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
#MCS	12	110	85	72	3	562	959	29	20	21	528	23	96	16	105	16	9
Time(s)	0.1	0.3	313.2	11.3	0.1	21236.	818.8	0.7	0.33	4.05	2572.1	0.23	11.2	8.33	283.38	0.77	0.95

210

211

212

213

214

VI. DIAGNOSTIC DES PANNES PAR SEQUENCE DE TESTS

Les méthodes de contrôle et de diagnostic du bon fonctionnement d'un système industriel ou autres permettent de mettre en place des stratégies de vérification de la bonne santé de leurs composants. Le fait que ces composants doivent être utilisés dans certaines conditions particulières, tels qu'un fonctionnement prolongé ou dans des environnements d'exploitation extrêmes; les propriétés de ces composants peuvent se dégrader au fil du temps et de l'utilisation vers un niveau de tolérance inacceptable, provoquant des défaillances plus régulières et même des arrêts problématiques en termes de sécurité et de perte d'argent, humaine et environnemental. Il est donc nécessaire de diagnostiquer les défauts avant un certain seuil visé et de permettre d'intervenir sur ces composants en procédant à des opérations de maintenance préventive ou simplement par remplacement par du neuf. La configuration d'éléments redondants est aussi une bonne solution pour permettre au système de continuer à fonctionner de manière satisfaisante sans pour autant provoquer son arrêt pour intervenir d'urgence sur les composant défaillants ou en voie de l'être.

Il existe un grand nombre de méthodes et techniques qui permettent de détecter et isoler les défaillances bien avant leur apparition. Celles-ci se répartissent en majorité en deux classes, l'une basée sur les modèles mathématiques telles que les statistiques et l'intelligence artificielle (machine Learning, deep Learning, réseaux de neurones artificielles) et d'autres plus pragmatiques telles que les pratiques sur site sur le matériel (inspection et maintenance), dans ce cas, une bonne équipe de techniciens en maintenance fait bien l'affaire. Les modèles mathématiques fournissent des méthodes d'estimation de l'information recueillie lorsque des défauts qui se produisent, et présentent des technologies d'apprentissage et de conception basées sur des modèles pour la détection, l'isolation et l'identification des pannes, ainsi que pour le contrôle tolérant aux pannes. Ces modèles sont également utilisés pour analyser les conditions de détectabilité des défauts et discuter de la stabilité du système en boucle fermée. Cependant, que ce soit l'une ou l'autre des méthodes de diagnostic, celles-ci requièrent du temps et surtout un coût qui pourrait être prohibitif, pour cela, le choix portera en priorité sur l'applicabilité de la méthode et son coût correspondant à sa mise en pratique. On revient dans ce cas à un problème d'optimisation qui consiste à trouver la série de coupes minimales constituées de composants responsable de la défaillance tout en minimisant le coût de tests. La procédure revient à établir en premier lieu un ordonnancement optimal des coupes minimales se basant sur le critère de minimisations de l'espérance mathématique du coût de tests. Les données utilisées sont le diagramme de fiabilité du système, la probabilité de défaillance des composants et leur coût de test.

239

240 Dans ce qui suit, nous proposons une méthode basée sur un modèle mathématique tiré d'un travail de notre équipe publiée
 241 dans Gao et Ait-Kadi (1994) et généralisée dans le mémoire de Master de Arous (2012) sous la direction de Daoud Ait-Kadi,
 242 co-auteur du présent article.

243 Supposons qu'un système soit formé de (m) coupes minimales mises en série comme montre la Figure 1.c. On cherche à
 244 trouver la coupe responsable de la défaillance tout en minimisant le coût de tests. Pour cela on procède comme suit :

245 Selon Gao et Ait-Kadi (1994) la séquence d'ordonnancement doit satisfaire les conditions suivantes :

$$\frac{p_j}{c_j} \geq \frac{p_{j+1}}{c_{j+1}}$$

$$c_{m-1} \leq c_m$$

248 Avec p_j est la probabilité conditionnelle que la coupe M_j soit responsable de la défaillance, et c_j est le coût de tests de la coupe
 249 M_j .

250 Cependant, Le coût total moyen de la séquence optimale de tests S_n^* est défini comme suit :

$$C(S_n^*) = E[C(S_n^*)] = c_1 + c_2(1 - p_1) + \dots + c_k \left(1 - \sum_{i=1}^{k-1} p_i \right) + c_{m-1} \left(1 - \sum_{i=1}^{m-2} p_i \right)$$

251 Avant de calculer le coût moyen, nous devons déterminer l'expression du coût total moyen de la séquence de tests des
 252 composants appartenant à la même coupe minimale M_j , Elle sera exprimée mathématiquement comme suit :

$$C(S_j) = c_{1j} + c_{2j}(p_{1j}) + \dots + c_{E_j j} \left(\prod_{k=1}^{E_j-1} p_{kj} \right)$$

260 (E_j) est le nombre de composants contenu dans la coupe M_j . Pour avoir la séquence optimale de tests qui minimise $C(S_j)$, il
 261 est utile de choisir en premier le composant le moins coûteux c'est-à-dire celui qui détient la plus grande probabilité d'être en
 262 bon état. Il est à rappeler que si une coupe minimale est responsable de la défaillance du système, cela veut dire que tous ces
 263 composants doivent être défaillants. Sinon, si elle n'est pas la cause de la panne, il faut vite trouver le composant qui est en bon
 264 état pour lâcher la coupe en question et passer à une autre.

265 Maintenant, selon GAO et Ait-Kadi (1994), pour chaque coupe, les composants doivent satisfaire la condition suivante :

$$\frac{c_{ij}}{1 - p_{ij}} \geq \frac{c_{ij+1}}{1 - p_{ij+1}}$$

266 Pour tout $i = 1, \dots, E_j - 1$

267 Notons que la probabilité de défaillance conditionnelle associée à une coupe minimale est déterminée par $p_k = \prod_{i \in \varphi} p_{ik}$ et son
 268 coût associé est égal à $c_k = \prod_{i \in \varphi} c_{ik}$, avec φ l'ensemble des indices des composants formant les coupes minimales.

269 La procédure suivante permet de fournir la séquence optimale de tests dans le cas d'indépendance des coupes minimales.
 270 Elle procède comme suit :

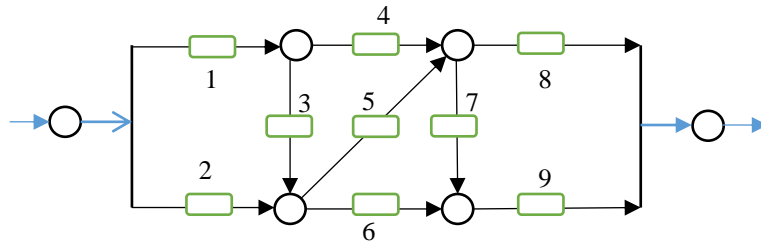
271 Procédure de génération de la séquence optimale de de tests

- 272 1- Déterminez les coupes minimales à partir du diagramme de fiabilité
- 273 1.1- Appliquez Procédure Génération de MCS
- 274 1.2- Appliquez la Procédure De_ Filtrage (CS, MCS)
- 275 2- Cherchez les p_j et les c_j de chaque coupe minimale
- 276 3- Classez les coupes selon l'ordre décroissant de p_j / c_j
- 277 4- Classez les composants suivant l'ordre croissant de $c_j / (1 - p_j)$

278 VII. APPLICATION À UN SYSTÈME DE PRODUCTION

279 Les modèles proposés, tels qu'ils ont été développés précédemment sont appliqués à une étude de cas industriel représentant
 280 un système de production composé de 9 machines reliées entre elles par 6 tampons (nœuds). Les tampons sont des composants
 281 idéaux du système, c'est pourquoi nous acceptons qu'ils ne tombent jamais en panne ou ne perturbent jamais le processus de
 282 production. Un tel type de composants a une fiabilité égale à 1. Cet exemple a été utilisé dans Rebaiaia et Ait-Kadi (2024) pour
 283 la maintenance opportuniste du système industriel.

292 Le système original était composé de 14 nœuds et 23 liens (lignes de production ou machines), et après avoir appliqué
 293 certaines règles de réduction (voir Rebaiaia, (2011), il a été réduit à un système à 6 nœuds et 9 liens, comme le montre la figure
 294 suivante :



295
296
297 Fig. 4. A 6-nodes and 23-links simplified system network

298 Après l'application des algorithmes qui permettent la génération des ensembles de coupes minimales tels que présentés
 299 auparavant nous obtenons ces coupes minimales telles que présentées dans la table suivante :

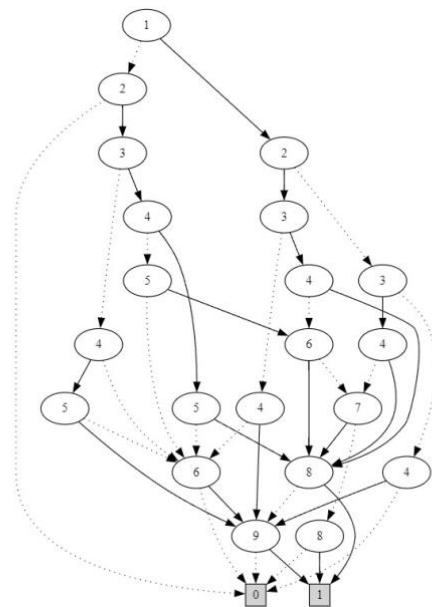
300
301 Table 2. L'ensemble des coupes minimales générée à partir du diagramme de fiabilité (Figure 4)

	# Item	1	2	3	4	5	6	7	8	9
Coupes Minimales	1	■	■							
	2	■				■	■			
	3	■				■				■
	4		■		■	■				
	5		■	■					■	
	6					■	■	■		
	7					■				■
	8							■		■
	9							■		■
	p_j	0.1	0.2	0.3	0.2	0.15	0.1	0.2	0.05	0.25
	c_j	10	10	25	15	20	35	15	30	20

302
303 Ces coupes minimales ont été déterminées à partir du ROBDD affiché dans la figure 5 :

```

304 digraph G {
0 [shape=box, label="0", style=filled, shape=box, height=0.3, width=0.3];
1 [shape=box, label="1", style=filled, shape=box, height=0.3, width=0.3]; 113 [label="1"]; 113 -> 108 [style=dotted]; 113 -> 112 [style=filled]; 108 [label="2"]; 108 -> 0 [style=dotted]; 108 -> 107 [style=filled]; 107 [label="3"]; 107 -> 102 [style=dotted];
107 -> 106 [style=filled]; 102 [label="4"]; 102 -> 99 [style=dotted]; 102 -> 101 [style=filled]; 99 [label="6"]; 99 -> 0 [style=dotted]; 99 -> 36 [style=filled]; 36 [label="9"]; 36 -> 0 [style=dotted]; 36 -> 1 [style=filled]; 101 [label="5"]; 101 -> 99 [style=dotted]; 101 -> 36 [style=filled]; 106 [label="4"]; 106 -> 104 [style=dotted]; 106 -> 105 [style=filled]; 104 [label="5"]; 104 -> 99 [style=dotted];
104 -> 103 [style=filled]; 103 [label="6"]; 103 -> 63 [style=dotted];
103 -> 62 [style=filled]; 63 [label="7"]; 63 -> 34 [style=dotted];
63 -> 62 [style=filled]; 34 [label="8"]; 34 -> 0 [style=dotted];
34 -> 1 [style=filled]; 62 [label="8"]; 62 -> 36 [style=dotted];
62 -> 1 [style=filled]; 105 [label="5"]; 105 -> 99 [style=dotted];
105 -> 62 [style=filled]; 112 [label="2"]; 112 -> 69 [style=dotted];
112 -> 111 [style=filled]; 69 [label="3"]; 69 -> 66 [style=dotted];
69 -> 68 [style=filled]; 66 [label="4"]; 66 -> 0 [style=dotted];
66 -> 36 [style=filled]; 68 [label="4"]; 68 -> 63 [style=dotted];
68 -> 62 [style=filled]; 111 [label="3"]; 111 -> 109 [style=dotted];
111 -> 110 [style=filled]; 109 [label="4"]; 109 -> 99 [style=dotted];
109 -> 36 [style=filled]; 110 [label="4"]; 110 -> 103 [style=dotted];
110 -> 62 [style=filled]; }
  
```



305
306 Figure 5. Code du graphe et ROBDD générés de la structure du système de production (Les MCS sont en pointillés).

307
308
309
310
311

L'arbre de défaillance relatif à ce cas d'étude contient 9 coupes minimales et autant de composants. Cependant, les probabilités de défaillance et les coûts de tests des composants sont affichés dans la table 3.

Table 3. Application de la procédure de génération des séquences de test

M_j	Composants de chaque coupe M_j	Probabilité de défaillance	Coût des tests	Le rapport $\frac{p_j}{c_j}$	Ordre	$r_j = \frac{c_j}{(1-p_j)}$
1	1, 2	0.02	20	0.001	2	$r_1 = 11,11, r_2 = 10.2$
2	1, 5, 6	0.0025	65	$38.46 \cdot 10^{-6}$	8	$r_1 = 11.11, r_5 = 25.53, r_6 = 38.89$
3	1, 5, 9	0,00375	50	$75 \cdot 10^{-6}$	5	$r_1 = 11.11, r_5 = 25.53, r_9 = 26.67$
4	2, 3, 4	0.012	50	$240 \cdot 10^{-6}$	3	$r_2 = 10.2, r_3 = 35.7, r_4 = 18.75$
5	2, 3, 7, 8	0.0006	80	$7.5 \cdot 10^{-6}$	9	$r_2 = 10.2, r_3 = 35.7, r_7 = 18.75, r_8 = 31.58$
6	4, 5, 6	0.003	70	$42.86 \cdot 10^{-6}$	7	$r_4 = 18.75, r_5 = 25.53, r_6 = 38.89$
7	4, 5, 9	0.0075	55	$136.4 \cdot 10^{-6}$	4	$r_4 = 18.75, r_5 = 25.53, r_9 = 23.53$
8	6, 7, 9	0.005	70	$71.43 \cdot 10^{-6}$	6	$r_6 = 38.89, r_7 = 18.75, r_9 = 23.53$
9	8, 9	0.125	50	0.0025	1	$r_8 = 31.58, r_9 = 23.53$

312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330

Nous avons classé les coupes minimales suivant le rapport p_j/c_j . Nous constatons d'après les calculs que l'ordre des coupes minimales est établi sur la colonne 6. Selon ces données le composant 9 est choisi pour être testé en premier. Si ce composant est en bon état de fonctionner d'après les calculs établis dans la colonne 7 après l'application de la formule $r_j = \frac{c_j}{(1-p_j)}$ alors les coupes minimales 3, 7, 8 et 9 sont écartées et donc il nous restera à tester que les composants des coupes 1, 2, 4, 5 et 6. Dans le cas contraire si le composant 9 est en défaut, on passe pour tester le composant 8. Si ce dernier est fonctionnel, on écarte les coupes minimales 5 et 9. On passe maintenant à la coupe minimale 1. On teste alors le composant 2. S'il est fonctionnel, on écarte les coupes minimales 1, 4, et 5. Si le composant 2 provoque la défaillance on passe au test du composant 1. Si ce dernier est bon, on écarte du test les coupes minimales 1, 2, et 3. Dans le cas contraire on passe à la coupe minimale 4. On teste d'abord le composant 2. Si ce dernier est en bon état, on écarte la coupe minimale 4. Dans le cas contraire, on passe au test du composant 3, s'il est bon, on écarte la coupe minimale 4. Puis on continue de proche en proche jusqu'à ce que toutes les coupes minimales soient considérées et que leurs composants soient testés.

Pour mieux représenter le cas d'application, considérons trois scénarios qui au moment du test nous supposons que des composants peuvent être en pannes représentés par la lettre P ou fonctionnels représentés par la lettre F. Ses scénarios sont donnés dans la table suivante :

Table 4. 3 scénarios proposés pour déterminer les séquences de test

Composants	Scénario A	Scénario B	Scénario C
1	F	F	F
2	D	F	F
3	F	D	F
4	D	D	D
5	F	F	D
6	D	F	D
7	F	D	F
8	D	D	F
9	F	F	F

331
332
333
334
335
336
337
338
339
340
341
342
343

Utilisons présentement les résultats contenus dans la table pour la suite des calculs.

Scénario A :

Coupe minimale 9 classée la première. Nous avons $\min(r_8 = 31.58, r_9 = 23.53) = r_9$ et donc c'est le composant 9 qui doit être vérifié. D'après la table colonne 1, le composant 9 est fonctionnel, on écarte alors les coupes minimale 7, 8 et 9. On considère maintenant la coupe minimale 1 qui est en deuxième position. Nous avons $\min(r_1 = 11,11, r_2 = 10.2) = r_2$. D'après la table colonne 1, le composant 2 est en panne, on vérifie alors le composant 1. Le composant 1 est fonctionnel, pour cela, on écarte les coupes minimales 1, 2, 3. On passe à la coupe minimale 4 pour tester les composants 2, 3 et 4. Nous avons $\min(r_2 = 10.2, r_3 = 35.7, r_4 = 18.75) = r_2$. Le composant 2 est défaillant, on passe alors au test du composant 4. Le composant 4 lui-même est défaillant, on passe au composant 3. Le composant 3 est fonctionnel, pour cela on écarte les coupes minimales 4 et 5. Comme on avait écarté toutes les coupes minimales, il nous reste une seule, c'est la 6 qui

est constituée par les composants 4, 5 et 6. D'après la table colonne 1, nous avons uniquement le composant 5 qui est fonctionnel. On arrête la procédure de test, les séquences de test ont été énumérées ci-dessus.

Nous pouvons continuer la même procédure de test en utilisant les scénarios B et C.

VIII. CONCLUSION

Dans le monde industriel, pour représenter un système de production on fait appel le plus souvent à une conceptualisation basée sur la théorie des graphes. En effet, tout système peut être modélisé sous forme de réseaux. Un réseau est constitué de nœuds et de liens qu'on appelle arcs ou arêtes, cela dépend de la propriété d'orientation. Ces réseaux sont le plus souvent très complexes. Les modéliser et les manipuler est un problème en lui-même NP-complexe. Pour un système multi-composant utilisant plusieurs technologies, il est souvent difficile de localiser le composant ou l'ensemble de composants responsable(s) d'une défaillance. Dans la plupart des cas c'est le technicien en maintenance qui procède par inspection du matériel déployé. Ceci n'est pas toujours possible devant la complexité du système qui nécessite l'automatisation du processus de diagnostic des composants sujets à des défauts ou à des bris incontrôlables. C'est dans cette mesure que notre article propose une solution efficace, facile à implanter et surtout économique. Cette solution est basée sur un modèle mathématique rigoureux et la théorie des graphes qui elle-même est un concept efficace et sûr. Nous avons appliqué cette méthode sur un cas industriel d'une unité de production. La méthode de calcul pour extraire les séquences de tests est basée sur un algorithme efficace, rapide et exact pour déterminer les coupes minimales qui sont essentielles pour le diagnostic et la détection des défaillances possibles. Nous avons utilisé un cas d'étude industriel connu de la littérature. Nous avons déterminé les MPS puis les MSC. Ces derniers ont permis de vérifier quelle coupe minimale est la plus fragile et qui est susceptible de provoquer la défaillance du système. Nous préconisons dans un future proche d'étendre cette méthode pour associer la théorie des ensembles flous telle que développée dans un précédent article publié dans Afshin Jamshidi et al. (2018).

BIBLIOGRAPHIE

- Afshin Jamshidi, Daoud Ait-kadi, Angel Ruiz, Mohamed Larbi Rebaiaia (2018), Dynamic risk assessment of complex systems using FCM," International Journal of Production Research, Taylor & Francis Journals, vol. 56(3), pages 1070-1088, February.
- Al-Ghanim AM. (1999), A heuristic technique for generating minimal path and cutsets of a general network. Computers & Industrial Engineering, 36, pp. 45:55, 1999.
- Ali Muhammad Rushdi, Motaz Hussain mashah (2021), Derivation of Minimal Cutsets from Minimal Pathsets for a Multi-State System and Utilization of Both Sets in Checking Reliability Expressions, June 2021 Journal of Engineering Research and Reports 20(8): pp. 22-33, DOI: 10.9734/JERR/2021/v20i817354
- Arous A., (2012), Stratégie de localisation du (des) Composants défaillants pour un système multi-Composant, Maitrise Département de Génie Mécanique/ Génie Industriel, Université Laval, Québec (QC), Canada.
- Bryant RE. (1992), Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. ACM Computing Surveys, 3, pp. 293:318, 1992.
- Gao J and Ait-Kadi (2014), Adaptive diagnostic procedure for failures in a series system, International Journal of Production Economics, 33 (1994) pp. 15-19.
- Giovanni Luca Amicucci, Fabio Pera, Andrea Tonti, (2023), Reliability analysis of nuclear instrumentation and control systems (Chapter 15), Instrumentation and Control Systems for Nuclear Power Plants, Woodhead Publishing Series in Energy, 2023, pp. 887-956
- Hardy G, Lucet C, N. Limnios. K-Terminal Network Reliability Measures with Binary Decision Diagrams, (2007). IEEE Transaction on Reliability, 56, pp. 506:515, 2007.
- Jasmon G. B, Kai O.S. (1985), A New Technique in Minimal Paths and Cuts set Evaluation. IEEE Transactions on Reliability, R34, pp. 136:143, 1985.
- jGRASP, Auburn University, <http://www.jgrasp.org> (accessed 10 May 2011).
- Kuo SY, Yeh., F.M, Lin Y.H. (2007), Efficient and Exact Reliability Evaluation for Networks with Imperfect Vertices. IEEE Transaction on Reliability, 56, pp. 288:300, 2007.
- Kvassay M, Rusnak P, Zaitseva E, Stanković RS. (2020), Multi-Valued decision diagrams in importance analysis based on minimal cut vectors. In 2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL) 2020; 9: pp. 265-270. IEEE
- Locks MO, Wilson JM. (1993), Note on disjoint products algorithms. IEEE Transactions on Reliability. 41: pp. 81-84, 1992.
- Rauzy, A. New Algorithms for fault tree analysis. Reliability Eng. and Systems Safety J; 40, pp. 203:210, 1993.
- Rebaiaia Mohamed-Larbi and Ait-Kadi, Daoud, (2010), Model based binary decision diagrams for complex networks reliability optimization, IFAC-PapersOnLine, Editor: Milik, Adam, Hryniewicz, Edward, Programmable Devices and Embedded Systems, Vol. 10 | Part. 1, IFAC-PapersOnLine.net, Elsevier, ISSN: 1474-6670, 2010.
- Rebaiaia Mohamed-Larbi, Daoud Ait-Kadi (2017), System Reliability Evaluation for Imperfect Networks Using Polygon-to-Chain Reduction, American Journal of Operations Research Vol.7 No.3, May 22, 2017, DOI: 10.4236/ajor.2017.73014
- Rebaiaia Mohamed-Larbi, Daoud Ait-Kadi and Arturo Merlano (2010), A methodology for modeling and evaluating the reliability of a radio-communication network, Advanced Maintenance Engineering, Vol. 1 | Part. 1, pp. 90-95, IFAC-PapersOnLine.net, Elsevier, ISSN: 1474-6670, 2010.

404 Rebaiaia Mohamed-larbi, Daoud Ait-Kadi and Arturo Merlano, (2009). A Practical Algorithm for network reliability evaluation
405 based on the factoring theorem — A case study of a generic Radiocommunication network, Journal of Quality, Vol. 16, N0 5,
406 pp. 323-335, November 2009
407 Rebaiaia Mohamed-larbi. (2011), A Contribution to the Evaluation and Optimization of Networks Reliability. PhD Thesis,
408 Laval University, Canada.
409 Rebaiaia Mohamed-larbi. and Ait-Kadi, D. (2015) Reliability Evaluation of Imperfect K-Terminal Stochastic Networks Using
410 Polygon-to-Chain and Series-Parallel Reductions. Proceedings of the 11th ACM Symposium on QoS and Security for Wireless
411 and Mobile Networks, Cancun, Mexico, 2-6 November 2015, pp. 115-122.
412 Rebaiaia Mohamed-Larbi & Ait-Kadi Daoud, (2024). Condition-based maintenance optimisation for multi-component systems
413 using mean residual life, International Journal of Production Research, Taylor & Francis Journals, vol. 62(13), pages 4831-
414 4855, July.
415 Rudell, R. Dynamic variable ordering for ordered binary decision diagrams. In: IEEE/ACM international conference on
416 Computer-aided design, pp. 42:47, 1993.
417 Shier DR, Whited DE. Algorithms for generating minimal cutsets by inversion. IEEE Transaction on Reliability, R-34, pp.
418 314:319, 1985.
419 Sunan Huang, Kok Kiong Tan, Poi Voon Er, and Tong Heng Lee (2020), Intelligent Fault Diagnosis and Accommodation
420 Control, CRC Press is an imprint of Taylor & Francis Group, LLC, 2020.
421 Yan L, Taha H, Landers L.A. (1994), Recursive Approach for Enumerating Minimal Cuts sets in a Network. IEEE Transactions
422 on Reliability, 43, pp. 383:388, 1994.
423 Yeh W-C. (2007). A Simple Heuristic Algorithm for Generating All Minimal Paths. IEEE Transactions on Reliability. 56: pp.
424 488-494, 2007.

425

426