



HAL
open science

MAINTENANCE DES SYSTÈMES DISTRIBUÉS : MÉTHODES D'AIDE À LA DÉCISION TEMPS-RÉEL

Kossi Pélopie Adzakpa

► **To cite this version:**

Kossi Pélopie Adzakpa. MAINTENANCE DES SYSTÈMES DISTRIBUÉS : MÉTHODES D'AIDE À LA DÉCISION TEMPS-RÉEL. Sciences de l'ingénieur [physics]. Université de Technologie de Troyes, 2004. Français. NNT: . tel-00371543

HAL Id: tel-00371543

<https://theses.hal.science/tel-00371543>

Submitted on 28 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MAINTENANCE DES SYSTÈMES DISTRIBUÉS : MÉTHODES D'AIDE À LA DÉCISION TEMPS-RÉEL

Auteur
Kossi Péloupé ADZAKPA

THÈSE PRÉSENTÉE POUR L'OBTENTION DU GRADE DE
DOCTEUR DE L'UNIVERSITÉ DE TECHNOLOGIE DE TROYES
TROYES, FRANCE
SOUTENUE LE 12 OCTOBRE 2004

SPÉCIALITÉ : OPTIMISATION ET SÛRETÉ DES SYSTÈMES

JURY :

Président :

Dimitris KIRITSIS

Dir. Scientifique Adjoint, École Polytechnique Fédérale de Lausanne

Rapporteurs :

Eric NIEL

Professeur des Universités, LAI - INSA de Lyon

Daniel NOYES

Professeur des Universités, LGP - ENI de Tarbes

Examineurs :

Kondo Hloindo ADJALLAH

MCF-HDR, Université de Technologie de Troyes, Directeur de thèse

Chengbin CHU

Professeur des Universités, Université de Technologie de Troyes

Zineb SIMEU-ABAZI

MCF-HDR, LAG - INP de Grenoble

UNIVERSITÉ DE TECHNOLOGIE DE TROYES
Département Génie des Systèmes Industriels
Institut des Sciences et Technologies de l'Information de Troyes - FRE CNRS 2732
Équipe d'Optimisation des Systèmes Industriels

Date : 12 Octobre 2004

UNIVERSITÉ DE TECHNOLOGIE DE TROYES

Date de soumission : **8 Juillet 2004**

Auteur : **Kossi Pélopie ADZAKPA**

Titre : **Maintenance des Systèmes Distribués : Méthodes d'Aide à la
Décision Temps-Réel**

Département : **Génie des Systèmes Industriels**

Grade : **Doctorat**

Soutenance : **12 Octobre**

Année : **2004**

L'auteur

À mon Père Seth
À ma Mère Grace
À mon Oncle Christian

Table des matières

Table des matières	ix
Liste des tableaux	xv
Liste des figures	xvii
Remerciements	xix
Résumé	xxi
Abstract	xxiii
1 Introduction générale	1
1.1 Introduction	1
1.2 La maintenance de ses débuts à nos jours	2
1.3 La maintenance dans l'entreprise : les différentes pratiques	4
1.3.1 Les domaines d'action de la fonction maintenance	4
1.3.2 Les différentes formes de maintenance	6
1.3.3 Objectifs visés par le « préventif »	6
1.3.4 Avantages et limites de la maintenance conditionnelle	6
1.3.5 La maintenance basée sur la fiabilité ou RCM	8
1.3.6 La maintenance productive totale ou TPM	9
1.3.7 La fonction ordonnancement en maintenance	11
1.4 Conclusion	12
2 La maintenance dans la recherche scientifique : état de l'art	15
2.1 Introduction	15
2.2 Critères de classification des problèmes de maintenance	16
2.3 Politiques d'inspection et maintenance conditionnelle	17
2.3.1 Politiques d'inspection	18

2.3.2	Remplacement et politique de maintenance conditionnelle	19
2.4	Réparation partielle et politiques de maintenance imparfaite	20
2.5	Problème de regroupement des activités de maintenance	21
2.6	Problèmes d'allocation de réparateurs	23
2.7	Planification de la maintenance et gestion de production	25
2.8	Conclusion	26
2.8.1	Principales observations de l'état de l'art	27
2.8.2	Objectifs de nos travaux	28
3	Maintenance avec coûts impliquant des temps de réponse	31
3.1	Introduction	31
3.2	Position et modélisation du problème	32
3.3	Configuration avec des entités non pondérées et des temps logistiques faibles	38
3.3.1	Le problème statique et les critères de décision	38
3.3.2	Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{11})	40
3.3.3	Protocole d'expérimentation	41
3.3.4	Résultats numériques	42
3.4	Configuration avec des entités pondérées et des temps logistiques faibles	45
3.4.1	Le problème statique et les règles de décision	46
3.4.2	Situation de la fonction <i>PRTWF</i> par rapport aux règles de dominance de la littérature	49
3.4.3	Algorithme de base	51
3.4.4	Exemples illustratifs de la procédure AlgoOF-PRTWF	52
3.4.5	Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{12})	54
3.4.6	Résultats numériques	56
3.5	Configuration avec des entités non pondérées et des temps logistiques considérables	56
3.5.1	Le problème statique et les règles de décision	58
3.5.2	Algorithmes de base	62
3.5.3	Borne inférieure du temps de réponse en présence de temps logistiques dépendant de la séquence	63
3.5.4	Exemples illustratifs des procédures avec des temps logistiques	67
3.5.5	Algorithmes en temps-réel pour la résolution du problème (\mathcal{P}_{13})	70
3.5.6	Résultats numériques	71
3.6	Configuration avec des entités pondérées et des temps logistiques considérables	73
3.6.1	Le problème statique et les règles de décision	74
3.6.2	Algorithme de base pour l'utilisation de la fonction PRTWFST	77

3.6.3	Borne inférieure du temps de réponse pondéré en présence de temps logistiques dépendant de la séquence	78
3.6.4	Illustration de la procédure avec des temps logistiques et des poids	82
3.6.5	Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{14})	84
3.6.6	Résultats numériques	86
3.7	Conclusion	87
4	Maintenance avec coûts impliquant des contraintes de délais	89
4.1	Introduction	89
4.2	Position du problème	90
4.2.1	Problème avec des coûts d'états critiques et de retards	90
4.2.2	Problème avec des coûts d'avances et retards	92
4.3	Coûts d'états critiques et retards avec des temps logistiques faibles	94
4.3.1	Le problème statique et les règles de décision	95
4.3.2	Algorithme de base	97
4.3.3	Borne inférieure du coût de temps de réponse et de retard dans le cas statique	98
4.3.4	Exemple 4.1 : illustration simple de la procédure AlgoOF-FTR	99
4.3.5	Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{21})	100
4.3.6	Protocole d'expérimentation	101
4.3.7	Résultats numériques	102
4.4	Coûts d'états critiques et retards avec des temps logistiques considérables	104
4.4.1	Le problème statique et les règles de décision	106
4.4.2	Algorithme de base	108
4.4.3	Exemple 4.2 : illustration de la procédure AlgoOF-FTRS	109
4.4.4	Algorithme temps-réel pour la résolution du problème (\mathcal{P}_{22})	111
4.4.5	Résultats numériques	112
4.5	Coûts d'avances et retards avec des temps logistiques faibles	114
4.5.1	Le problème statique et les règles de décision	114
4.5.2	Algorithme de base	116
4.5.3	Exemple 4.3 illustrant la procédure AlgoOF-ETR	117
4.5.4	Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{31})	118
4.5.5	Résultats numériques	119
4.6	Coûts des avances et retards avec des temps logistiques importants	121
4.6.1	Le problème statique et les règles de décision	121
4.6.2	Algorithme de base	122
4.6.3	Exemple 4.4 illustratif de la procédure AlgoOF-ETRS	123

4.6.4	Algorithme en temps-réel du problème (\mathcal{P}_{32})	124
4.6.5	Résultats numériques	126
4.7	Conclusion	126
5	Organisation de la Maintenance et Simulation Monte-Carlo	129
5.1	Introduction	129
5.2	Position du problème	129
5.3	L'outil de simulation Monte-Carlo	132
5.4	Algorithmes de simulation pour les problèmes \mathcal{P}_1	132
5.4.1	Simulation pour le problème \mathcal{P}_{11}	132
5.4.2	Simulation pour le problème \mathcal{P}_{12}	133
5.4.3	Simulation pour le problème \mathcal{P}_{13}	134
5.4.4	Simulation pour le problème \mathcal{P}_{14}	136
5.5	Algorithmes de simulation pour les problèmes \mathcal{P}_2	137
5.5.1	Simulation pour le problème \mathcal{P}_{21}	137
5.5.2	Simulation pour le problème \mathcal{P}_{22}	138
5.6	Algorithmes de simulation pour les problèmes \mathcal{P}_3	139
5.6.1	Simulation pour le problème \mathcal{P}_{31}	139
5.6.2	Simulation pour le problème \mathcal{P}_{32}	140
5.7	Résultats numériques	141
5.7.1	Protocole d'expérimentation	141
5.7.2	Résultats du problème \mathcal{P}_{11}	142
5.7.3	Résultats du problème \mathcal{P}_{12}	142
5.7.4	Résultats du problème \mathcal{P}_{13}	142
5.7.5	Résultats du problème \mathcal{P}_{14}	142
5.7.6	Résultats du problème \mathcal{P}_{21}	147
5.7.7	Résultats du problème \mathcal{P}_{22}	147
5.7.8	Résultats du problème \mathcal{P}_{31}	147
5.7.9	Résultats du problème \mathcal{P}_{32}	147
5.8	Observations et conclusion	152
6	Conclusion générale et perspectives	153
6.1	Conclusions	153
6.2	Perspectives	155
6.2.1	Problèmes avec des ressources de différentes classes	155
6.2.2	Maintenance avec fenêtres temporelles imposées	155

6.2.3	Étude de caractéristiques asymptotiques dans les politiques proposées . . .	156
6.2.4	Méthodes exactes avec les outils développés	156

Appendices **159**

A Concepts de base **159**

A.1	Introduction	159
A.2	Définitions	159
A.2.1	Système	159
A.2.2	Défaillance	159
A.2.3	Panne	160
A.3	Sûreté de fonctionnement	160
A.3.1	Fiabilité	160
A.3.2	Sécurité	161
A.4	Disponibilité	161
A.4.1	Disponibilité de deux composants en série	162
A.4.2	Disponibilité de deux composants en parallèle	162
A.4.3	Expression de la disponibilité	162
A.5	Maintenabilité et maintenance	163
A.6	Taux de défaillance, de réparation et de remise en service	164
A.6.1	Taux de défaillance	164
A.6.2	Taux de réparation	164
A.6.3	Taux de remise en service	165
A.7	Autres définitions	165
A.7.1	Le MTTF (Mean Time To Failure)	165
A.7.2	Le MTTR (Mean Time to Repair) ou (Mean Time To Restoration)	166
A.7.3	Et aussi...	166

B La disponibilité dans les cas non exponentiels **167**

B.1	Introduction	167
B.2	La fonction disponibilité dans le cas général	167
B.2.1	Équation de la fonction disponibilité dans le cas général	168
B.2.2	Expression de la fonction disponibilité dans le cas général	168
B.3	Disponibilité dans le cas d'une loi exponentielle	170
B.3.1	Équation de la disponibilité dans le cas exponentiel	170
B.3.2	Expression de la disponibilité dans le cas exponentiel	171

B.4	Caractéristiques de la loi de Weibull	172
B.4.1	Rappels et définitions	172
B.4.2	Étude du taux de défaillance $\lambda(t)$ suivant la loi de Weibull	172
B.5	La fonction Disponibilité dans le cas d'une loi de Weibull	175
B.5.1	Loi de Weibull et taux de réparation général	175
B.5.2	Loi de Weibull et taux de réparation constant	177
B.6	Conclusion	194

Bibliographie	197
----------------------	------------

Liste des tableaux

3.1	Instance de l'exemple 3.1 pour la fonction PRTWF	52
3.2	Valeurs de la fonction PRTWF à $t = 0$ dans l'exemple 3.1	53
3.3	Matrice de dominance à $t = 0$ dans l'exemple 3.1	53
3.4	Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.1	53
3.5	Autres décisions dans l'exemple 3.1 pour la fonction PRTWF	53
3.6	Séquence d'exécution des tâches de l'exemple 3.1	53
3.7	Instance de l'exemple 3.2 pour la fonction PRTWF	54
3.8	Séquence d'exécution des tâches de l'exemple 3.2	54
3.9	Instance de l'exemple 3.3 pour les fonctions PRTFST et PRTFSS	68
3.10	Matrice des temps logistiques s_{ij} dans l'exemple 3.3	68
3.11	Valeurs de la fonction PRTFST à $t = 0$ dans l'exemple 3.3	68
3.12	Matrice de dominance à $t = 0$ dans l'exemple 3.3	68
3.13	Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.3	69
3.14	Autres décisions dans l'exemple 3.3 pour la fonction PRTFST	69
3.15	Séquence d'exécution des tâches de l'exemple 3.3	69
3.16	Séquence d'exécution des tâches dans la borne inférieure de l'exemple 3.3	69
3.17	Matrice des temps logistiques s_{ij} dans l'exemple 3.4	70
3.18	Valeurs de la fonction PRTFSS des tâches à $t = 0$ dans l'exemple 3.4	70
3.19	Autres décisions dans l'exemple 3.4 pour la fonction PRTFSS	70
3.20	Instance de l'exemple 3.5 pour la fonction PRTWFST	83
3.21	Matrice des temps logistiques s_{ij} dans l'exemple 3.5	83
3.22	Valeurs de la fonction PRTWFST à $t = 0$ dans l'exemple 3.5	83
3.23	Matrice de dominance à $t = 0$ dans l'exemple 3.5	83
3.24	Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.5	84
3.25	Autres décisions dans l'exemple 3.5 pour la fonction PRTWFST	84
3.26	Séquence d'exécution des tâches de l'exemple 3.5	84
3.27	Séquence d'exécution des tâches dans la borne inférieure de l'exemple 3.5	84

4.1	Instance de l'exemple 4.1 pour les fonctions FTR et FTRW	99
4.2	Valeurs de la fonction FTR à $t = 0$ dans l'exemple 4.1	99
4.3	Matrice de dominance à $t = 0$ dans l'exemple 4.1	99
4.4	Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.1	99
4.5	Séquence d'exécution des tâches de l'exemple 4.1	100
4.6	Instance de l'exemple 4.2 pour les fonctions FTRS et FTRSW	109
4.7	Matrice des temps logistiques s_{ij} dans l'exemple 4.2 avec la fonction FTRS	110
4.8	Valeurs de la fonction FTRS à $t = 0$ dans l'exemple 4.2	110
4.9	Matrice de dominance à $t = 0$ dans l'exemple 4.2	110
4.10	Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.2	110
4.11	Autres décisions dans l'exemple 4.2 pour la fonction FTRS	111
4.12	Séquence d'exécution des tâches de l'exemple 4.2	111
4.13	Valeurs de la fonction ETR à $t = 0$ dans l'exemple 4.3	117
4.14	Matrice de dominance à $t = 0$ dans l'exemple 4.3	117
4.15	Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.3	117
4.16	Séquence d'exécution des tâches de l'exemple 4.3 avec la fonction ETR	118
4.17	Valeurs de la fonction ETRS à $t = 0$ dans l'exemple 4.4	124
4.18	Matrice de dominance à $t = 0$ dans l'exemple 4.4	124
4.19	Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.4	124
4.20	Autres décisions dans l'exemple 4.4 pour la fonction ETRS	124
4.21	Séquence d'exécution des tâches de l'exemple 4.4	124

Liste des figures

1.1	Modèle entrée-sortie de la maintenance dans l'entreprise	4
1.2	Principaux domaines d'action de la fonction maintenance	5
1.3	Graphe des différentes formes de maintenance	7
3.1	Structure organisationnelle d'un groupe à soutenir par des opérations de maintenance . . .	32
3.2	Définition de $\Delta_{ik}(t)$	40
3.3	Distribution des données utilisées du système à maintenir	43
3.4	Résultats avec des poids égaux et des temps logistiques faibles	44
3.5	Résultats avec des poids inégaux et des temps logistiques faibles	57
3.6	Différence entre π_1 et π_2 avec temps logistiques	65
3.7	Résultats avec des poids égaux et des temps logistiques importants	73
3.8	Les différentes règles pour la minimisation du temps de réponse	77
3.9	Différence entre π_3 et π_4 avec temps logistiques et poids	80
3.10	Résultats avec des poids inégaux et des temps logistiques importants	86
4.1	Distribution des données utilisées du système à maintenir avec les délais	103
4.2	Résultats des coûts d'états critiques et de retards avec des temps logistiques faibles	105
4.3	Résultats des coûts d'états critiques et de retards avec des temps logistiques im- portants	113
4.4	Résultats des coûts d'avance et de retards avec des temps logistiques faibles	120
4.5	Résultats des coûts d'avance et de retards avec des temps logistiques importants . .	127
5.1	Résultats de simulation pour le problème \mathcal{P}_{11}	143
5.2	Résultats de simulation pour le problème \mathcal{P}_{12}	144
5.3	Résultats de simulation pour le problème \mathcal{P}_{13}	145
5.4	Résultats de simulation pour le problème \mathcal{P}_{14}	146
5.5	Résultats de simulation pour le problème \mathcal{P}_{21}	148
5.6	Résultats de simulation pour le problème \mathcal{P}_{22}	149
5.7	Résultats de simulation pour le problème \mathcal{P}_{31}	150

5.8	Résultats de simulation pour le problème \mathcal{P}_{32}	151
A.1	Défaut, défaillance et panne d'un système	161
A.2	Grandeurs moyennes de sûreté de fonctionnement	166
B.1	Évolution du taux de défaillance suivant les valeurs du paramètre de forme	175
B.2	Évolution de la disponibilité suivant les valeurs du paramètre de forme	190
B.3	Fonctions disponibilité limites A_0 et A_∞	192

Remerciements

Cette thèse a été réalisée à l'Université de Technologie de Troyes au sein de l'Équipe d'Optimisation des Systèmes Industriels de l'Institut des Sciences et Technologies de l'Information de Troyes. Je tiens à remercier les dirigeants de cette Université de m'avoir accueilli et soutenu dans la réalisation de ce travail. Je remercie tout particulièrement le Professeur Chengbin CHU, Directeur de l'équipe OSI de l'ISTIT qui me fait en plus l'honneur d'examiner mon travail de thèse.

Ce travail n'aurait pas été possible sans le présence sans faille de Kondo H. ADJALLAH qui, plus qu'un directeur de thèse, a été un frère pour moi. Il a su canaliser les efforts que nous avons consentis ensemble et m'a guidé efficacement dans les directions de recherche suivies. Je te dis, Kondo, un grand merci du fond de mon cœur pour ton soutien moral qui m'a été d'un grand réconfort depuis mon premier jour à Troyes. Trouves ici ma profonde gratitude.

Je suis extrêmement reconnaissant à Monsieur Eric NIEL, Professeur des Universités à l'INSA de Lyon et à Monsieur Daniel NOYES, Professeur des Universités à l'ENI de Tarbes de me faire l'honneur d'être rapporteurs de ce travail.

Je remercie Monsieur Dimitris KIRITSIS, Directeur Scientifique Adjoint au LICP (Laboratoire des outils Informatiques pour la Conception et la Production) à l'École Polytechnique Fédérale de Lausanne et Madame Zineb SIMEU-ABAZI, Maître de Conférence - HDR à l'INP de Grenoble de consacrer une partie de leur temps pour examiner mon travail de thèse.

Si j'ai eu la force et surtout le moral au cours de ces quatre années passées ici, avec des journées de travail très longues, c'est grâce à l'environnement très convivial entretenu dans le département Génie des Systèmes Industriels. Je remercie Christian PRINS, Directeur de ce département et tous les membres du département pour cet environnement. Mes pensées vont particulièrement à mes collègues de bureau Saïd, Ahmed et Djamel et à Aly, Salif et Boubacar.

Ma gratitude va aux camarades et amis des autres laboratoires avec qui j'ai passé d'agréables moments. Je pense en particulier à Pierre avec qui les moments de vélo dans la campagne me soulageaient du stress du travail, et à tous les amis d'Ellidoc.

Et, non des moindres, je dis un grand merci à Sika de m'avoir apporté un soutien moral et d'avoir été une famille pour moi depuis mon premier jour à Troyes. Saches Sika que je ne saurai te remercier assez.

À mes parents, frères, sœurs et à toute ma famille, je ne pourrai dire assez merci pour tout ce qu'ils sont et font pour moi. Trouvez ici toute la pensée du fond de mon cœur.

Loin de moi l'envie d'oublier tous ceux qui me sont très chers et à qui je dois des dettes de reconnaissance mais que l'économie des mots m'oblige à ne pas citer ici. Je vous dis à tous un grand merci et vous dédie ce travail.

À tous, je dis : Merci! Merci! Vraiment merci du fond de mon cœur!

Pelope

Résumé

Au cours de ces dernières décennies, les systèmes technologiques ont beaucoup évolué. Dans le même temps, leurs installations sont de plus en plus distribuées sur plusieurs sites. Ceci conduit à la résolution de problèmes logistiques lors des prises de décision en vue d'actions de maintenance qui exigent par ailleurs des approches coopératives dans un contexte de gestion à distance.

Dans ce travail, nous étudions des méthodes d'aides à la décision temps-réel pour la maintenance des systèmes multi-sites distribués dans un environnement à ressources partagées, avec des contraintes logistiques non négligeables entre les différents sites. Les exigences sont de planifier et d'ordonnancer les tâches de maintenance, de les affecter en temps-réel aux ressources de maintenance disponibles, tout en minimisant les coûts inhérents au fonctionnement et en maîtrisant les délais d'intervention des ressources. Les coûts sont relatifs notamment au fonctionnement en état dégradé (état critique), aux fréquences trop élevées ou au déficit de maintenance. Ils dépendent aussi de combinaisons linéaires ou convexes d'ensemble de paramètres tels que les temps de réponse (temps de séjour), les retards et avances ainsi que des facteurs de pondération des tâches de maintenance. Les coûts sont également fonction de la disponibilité, cette dernière étant révélatrice des dégradations des entités constituant le système.

Pour satisfaire ces exigences, différents problèmes doivent être résolus dans le but de maintenir la disponibilité du système au dessus d'une limite minimale, de minimiser le coût des opérations de maintenance et d'affecter en temps-réel les tâches aux ressources de maintenance, le tout dans un processus optimal de décision. Ce processus doit tenir compte de différentes contraintes : dates d'arrivée inégales des tâches, temps logistiques non négligeables entre les sites, séquences optimales des tâches d'une intervention, précédences entre les tâches d'une même entité et critères de priorité et d'urgence de chaque tâche. Dans cette démarche, les caractéristiques et lois de comportement en fiabilité et en maintenabilité, et les sollicitations d'intervention sur les entités sont connues.

Pour résoudre ces problèmes, nous proposons une approche mettant en œuvre des méthodes d'ordonnement d'activités (tâches ou actions) de maintenance basées sur des règles de priorité dont nous montrons l'optimalité locale (par rapport au temps) dans ce document. Partant d'une ressource de maintenance, ces règles de décision s'inspirent des principes d'ordonnement sur une seule machine. Elles sont utilisées dans des algorithmes d'aide à la décision en temps-réel pour la planification dynamique des tâches de maintenance des systèmes distribués. Aux solutions fournies par la plupart de ces règles nous avons déterminé des bornes inférieures.

Nous avons structuré ce document en six chapitres dont le premier introduit de façon générale la maintenance et ses différentes pratiques dans les entreprises. Le chapitre 2 propose une analyse de l'état de l'art des travaux de recherche scientifique dans le domaine de la gestion des activités de maintenance. Ensuite, dans le chapitre 3, des approches de résolution des problèmes relatifs aux coûts des états critiques du système sont proposées. Le chapitre 4 traite alors les différents

problèmes relatifs aux contraintes de délais sur les opérations de maintenance induisant notamment des coûts combinés liés aux états critiques et aux retards d'intervention d'une part, et des coûts dus aux avances ou retards des tâches d'autre part. Il apparut alors nécessaire d'étudier la robustesse des performances des approches proposées dans les chapitres 3 et 4 en environnements incertains. Cette étude fût donc réalisée dans le chapitre 5 par des simulations Monte Carlo de la maintenance à partir des données d'un système réel soumis à des aléas de défaillance appelant des actions correctives. Enfin, de l'ensemble des investigations menées dans ce travail furent tirées les principales conclusions et les perspectives proposées dans le chapitre 6.

Mots clés : systèmes distribués, disponibilité, maintenance automatisée, ordonnancement, système d'aide à la décision, temps-réel.

Abstract

The technological systems have greatly evolved during the last past decades. In the same time, their installations are more and more scattered on different distributed sites. This emphasized the increasing concern about the maintenance decision making processes where cooperative approaches are required, especially in distributed systems.

In this thesis, we study methods for real-time decision making in the maintenance of large distributed multisite systems with important logistic constraints (set-up times), where the available maintenance resources are shared between the different sites in the system. The aims are to plan and schedule the maintenance tasks and assign the resources in real-time to them, while minimizing the costs that are involved. The different costs occur namely when the system operates in degradation states, when the maintenance is too often undertaken, or in case of insufficient maintenance. These costs depend on different parameters namely the weighted or unweighted response time (flow-time) of the maintenance tasks, their tardiness and their earliness. The costs also introduce linear and convex combinations of these different parameters and are generally based on the availability function which reveals the degradation state of the different assets in the system.

To reach the targets, different problems ought to be solved in order to guarantee a minimal required availability to the system, and assign in real-time the resources to the maintenance tasks. This must take into account the different constraints on the maintenance tasks which have unequal release dates and occur in real-time in the system. The set-up times between these tasks are important and depend on their sequence. Moreover, the tasks are linked by precedence constraints and are characterized by their priority and their emergency which are of a great importance in maintenance.

The resolution methods that we propose are based on scheduling approaches, with local optimality rules (depending on the time) that we prove in this document. The decision making rules consider, in a first time, a single maintenance resource with a link to single machine scheduling problems. We developed lower bounds for the off-line use of the rules, before adapting them in algorithms for real-time decision making in order to help the maintenance manager in the cooperative maintenance planning in distributed systems. The main goal is to minimize the different costs while controlling the intervention delay of the maintenance resources.

This document contains 6 main chapters. The first one is a general introduction to maintenance and presents the different practices of maintenance in the industry. In the second chapter, the main aspects of maintenance in the scientific research are analyzed. Then we propose in chapter 3 different approaches to solve the problem relative to the critical states costs of the system, which correspond to the maintenance tasks' response-time. We develop, in chapter 4, approaches to handle the different costs which are due to deadlines on the maintenance actions. In this chapter, we consider namely the combined cost of the critical states and the maintenance tasks' tardiness on the one hand, and the costs due to the tasks' earliness and tardiness on the other hand. In order to analyze the performances of the different approaches developed in chapter 3 and chapter 4, and study the

robustness of these approaches in a random environment, we simulated them in chapter 5 with uncertainties. Finally we present in chapter 6 the main conclusions of the document and future extensions to this work.

Key words: Distributed systems, availability, maintenance, scheduling, real-time decision making.

Chapitre 1

Introduction générale

1.1 Introduction

A l'origine destinée à rétablir « maintenir » les machines militaires dans un état où elles ont un fonctionnement optimal, la maintenance s'est vite répandue à tous les niveaux de la technologie. Elle a dorénavant dépassé le simple cadre des machines-outils pour couvrir les domaines de tous les biens humains dégradables, s'étendant des puces électroniques jusqu'aux grands édifices publics. Les études de maintenance et maintenabilité, tout comme tous les autres domaines de la sûreté de fonctionnement, science dont elle fait partie, sont présentes depuis la conception jusqu'à l'installation, l'exploitation et la mise au rebut de tous les systèmes technologiques. À l'image de son évolution et de la variété de ses applications, la maintenance est constituée d'activités variées qui vont du simple nettoyage des composants jusqu'à des activités plus complexes voire jusqu'à leur modification et remplacement.

Pour rester compétitives, les entreprises concurrentes sont obligées de chercher à faire au moins aussi bien que les autres. Une politique judicieuse de maintenance peut y contribuer efficacement. La fonction maintenance constitue une nécessité. Elle se révèle en effet être indispensable pour assurer la disponibilité des équipements. Si nous nous permettons de paraphraser Pierre Dac, "mieux vaut penser le changement que changer le pansement". La fonction maintenance présente également une grande importance pour la sécurité des personnes et des biens. Outre son importance pour la sécurité des personnes et des biens et sa place dans la qualité des produits finis, la maintenance est nécessaire pour la durée de vie des outils et des installations.

Ainsi, comme l'a relevé Tsang [153], l'environnement économique actuel a élevé à un niveau d'importance stratégique la fonction maintenance au sein des organisations, surtout celles qui ont un important investissement dans les entités physiques. La maintenance est souvent regardée comme une dépense nécessaire qui fait partie des budgets de fonctionnement. La disponibilité et la fiabilité des entités étant devenues des aspects cruciaux des organisations, ceci a influé de façon non négligeable sur la maintenance.

Un management judicieux de la maintenance est un moyen incontournable pour réduire les

coûts et accroître la compétitivité des entreprises. Fonction indispensable et d'une grande importance stratégique dans les entreprises fournissant des biens et services, la maintenance vit présentement une véritable mutation, tant sur les plans de l'organisation humaine que technologique [32].

Dans les industries fortement automatisées, le personnel de maintenance représente souvent de l'ordre de 40% de l'effectif total. Toute-fois, le coût direct de mise en œuvre de la maintenance ne constitue que l'un des facteurs à considérer. Un autre facteur économique primordial réside dans le coût que constituent les coûts d'indisponibilité ou de dégradation de fonctions d'équipements. Ce dernier coût mérite qu'on lui porte une grande attention, même plus que le coût direct, étant donné que ce dernier est un coût qu'on doit considérer a priori.

Le reste de ce chapitre sera organisé comme suit. Dans la section 2, nous présenterons très brièvement l'évolution des pratiques de la maintenance depuis son origine. Elle sera suivie dans la section 3 par une présentation générale des différentes pratiques de maintenance en cours dans les entreprises. La conclusion de ce chapitre viendra alors faire un point sur cette évolution de la maintenance et sa pratique et présentera les grandes lignes de ce document.

1.2 La maintenance de ses débuts à nos jours

Bien avant la révolution industrielle dans la seconde moitié du dix-huitième (18^e) siècle, la maintenance consistait en des activités des artisans tels que les menuisiers, les forgerons, les soudeurs, les maçons, etc, dans la réparation des bâtiments, des machines primitives, et des moyens de transport de l'époque [141]. Comme il n'y avait pas de concepts tel que « pièces de rechange » ou autre concepts relatifs à la maintenance de jours, les défaillances étaient en général réparées par la fabrication de nouvelles pièces ou la réparation de l'ancienne. Par ailleurs, la réparation était très à l'ordre du jour et les structures de base étaient elles mêmes soit réparables, soit d'une très grande durabilité, ou les deux à la fois. En l'absence de méthodes pour le calcul de la fatigue et autres concepts liés à la dégradation, la conception et la réparation était fortement intégrées, en ce sens que pour remplacer une pièce défaillante jugée pas assez efficace par le propriétaire, ce dernier exigeait une autre beaucoup plus « robuste ». Toute ces pratiques étaient liées à l'évolution lente et au faible coût de ces services de l'époque.

Ces pratiques ont progressivement évolué à partir de la révolution industrielle. En 1785, Thomas Jefferson a noté que les pièces des canons étaient fabriquées de façon adéquate et assez rapidement pour qu'on envisage de les changer au besoin [141]. Le concept, lent à l'époque, de pièces de rechange, s'est progressivement accéléré. Cette accélération est fortement accentuée par la complexité croissante des machines à « maintenir ». Le travail de l'agent de maintenance s'est alors progressivement dirigé vers plus d'aptitudes au diagnostic. Un autre facteur important de ce changement était le contrôle de qualité, puis l'automatisation dans les systèmes manufacturiers ; ceci a rendu le remplacement des pièces, et plus tard des unités entières, plus rapide et plus facile que les réparations dans beaucoup de cas.

Les approches pour la gestion de la maintenance ont ainsi considérablement évolué au cours du vingtième siècle. Jusqu'en 1940, seule la maintenance corrective était pratiquée et elle était considérée comme un coût inévitable [116]. Les agents spécialisés de maintenance (réparateurs) sont là pour la remise en état des équipements seulement une fois que ceux-ci sont tombés en panne. La maintenance n'était pas envisagée à la conception, et ses impacts sur les performances des systèmes n'étaient pas non plus reconnus.

Les applications de la recherche opérationnelle à l'industrie (notamment l'industrie militaire) pendant la deuxième guerre mondiale ont conduit à une large pratique de la maintenance préventive [163], [164]. Les Allemands, dans leurs efforts de guerre étaient de ceux qui ont fortement poussé à l'exigence de plus en plus de fiabilité, et partant accordent de plus en plus d'importance stratégique à la maintenance. La maintenance périodique était initialement destinée à améliorer la sûreté plutôt qu'à accroître la disponibilité ou à réduire les coûts.

Depuis les années 1950, les modèles de recherche opérationnelle sont apparus de façon croissante dans l'étude et l'optimisation des politiques de maintenance. Cette évolution peut être observée à travers les nombreux travaux publiés sur la maintenance et qui sont regroupés et analysés dans plusieurs travaux de synthèse et états de l'art tels que [19], [38], [49], [52], [56], [57], [73], [86], [107], [109], [125], [126], [135], [140], [141], [151], [156], [165] et [168]. Ces différents travaux, sur lesquels nous reviendrons, présentent des modèles qui traitent des effets de différentes politiques de maintenance et de la sélection optimale des paramètres de ces politiques. Mais l'impact réel des actions de maintenance sur la performance des activités n'est abordé que dans les années 1970 [116] où des approches plus intégrées de la maintenance ont évolué aussi bien dans le domaine publique que privé. L'acquisition de moyens de défense de plus en plus coûteux conduisit alors les autorités américaines à requérir des approches de « coût globale du cycle de vie » (LCC - life cycle costing), le coût de la maintenance devenant alors un composant très important de ce coût du cycle de vie. Le lien direct entre la fiabilité et la maintenabilité est alors reconnu et le terme "R&M" (Reliability-Maintenability) devient très largement utilisé en parlant des systèmes de défense. Ce concept est aussi adopté dans les industries et services civiles. La maintenance basée sur la fiabilité (RCM - Reliability Centered Maintenance) apparaît alors dans l'aviation civile aux États-Unis. Au même moment, les japonais développent la maintenance productive totale (TPM - Total Productive Maintenance) [150]. Ces deux pratiques, sur lesquelles nous reviendrons plus tard, considèrent la maintenance dans le contexte plus large des activités économiques et prennent en compte les liens étroits entre les défaillances des composants et leurs impacts sur la performance des activités. Ces deux pratiques opèrent sur des horizons à court et moyen terme. Les considérations sur ces horizons à court et moyen terme nécessitent des modèles de prédiction efficaces pour accéder aux conditions de divers éléments des systèmes et à leur comportement sous différentes stratégies de maintenance. Comme l'ont relevé Murthy *et al.* dans [116], des insuffisances dans ces pratiques sont devenues apparentes dans les années 1990. Il apparut de plus en plus que la maintenance ne devrait pas être vue que dans un contexte opérationnelle en considérant les défaillances des équipements et leurs conséquences. Elle devait tenir compte aussi bien des retombées économiques que des normes nationales et internationales de plus en plus exigeantes.

Ainsi, l'évolution de la maintenance et des outils pour son management ont donné à la maintenance diverses pratiques au sein de l'entreprise.

1.3 La maintenance dans l'entreprise : les différentes pratiques

Comme relevé dans la section précédente, la pratique de la maintenance qui, au départ était exclusivement de la réparation (maintenance curative), a évolué de façon drastique et les pratiques sont actuellement très diversifiées dans l'entreprise. La maintenance constitue actuellement une partie toute entière de l'entreprise avec des besoins (pour sa réalisation) et des retombées (résultats attendus). Ainsi, la maintenance au sein de l'entreprise se retrouve comme sur la figure 1.1 comme spécifiée par Tsang [153]

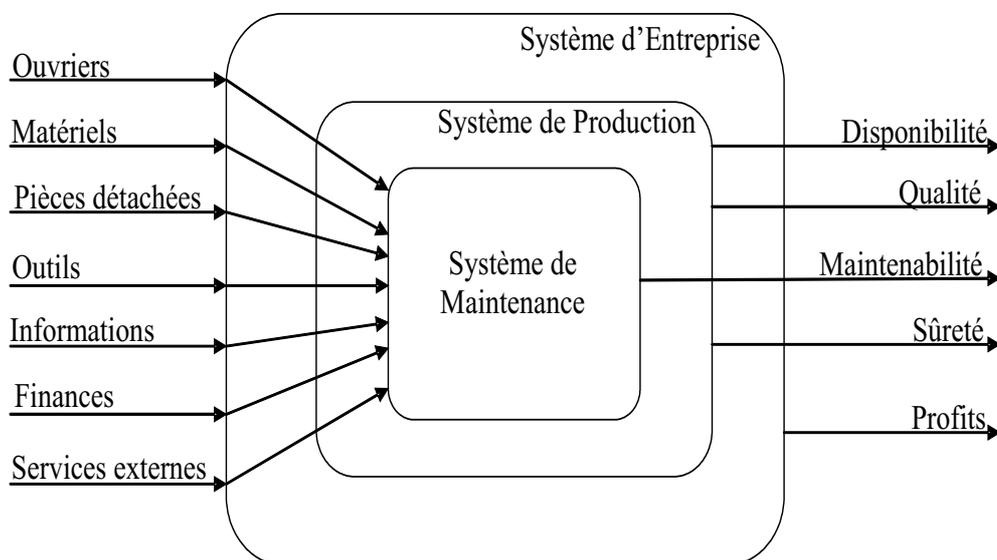


Figure 1.1 – *Modèle entrée-sortie de la maintenance dans l'entreprise*

1.3.1 Les domaines d'action de la fonction maintenance

Les activités rentrant dans le cadre de la maintenance sont au moins aussi diverses que les pratiques, et comme l'ont souligné Bouche *et al.* [30], les implications de la fonction maintenance peuvent être regroupées comme sur la figure 1.2 qui est très explicite en regard des activités relatives à ce domaine. Ainsi, la fonction maintenance est devenue, avec le temps, un domaine complexe où interviennent différentes compétences parmi lesquelles on trouve les méthodes de recherche opérationnelle pour l'optimisation de divers aspects : l'optimisation des coûts, la gestion des stocks des pièces de rechanges, la planification et l'ordonnancement des interventions compte tenu des capacités d'action de l'entreprises ; les compétences pour le diagnostic, pour la prédiction afin d'accomplir les actions de maintenance préventive, etc.

L'évolution de la maintenance a donc ainsi entraîné l'apparition de divers problèmes qui lui

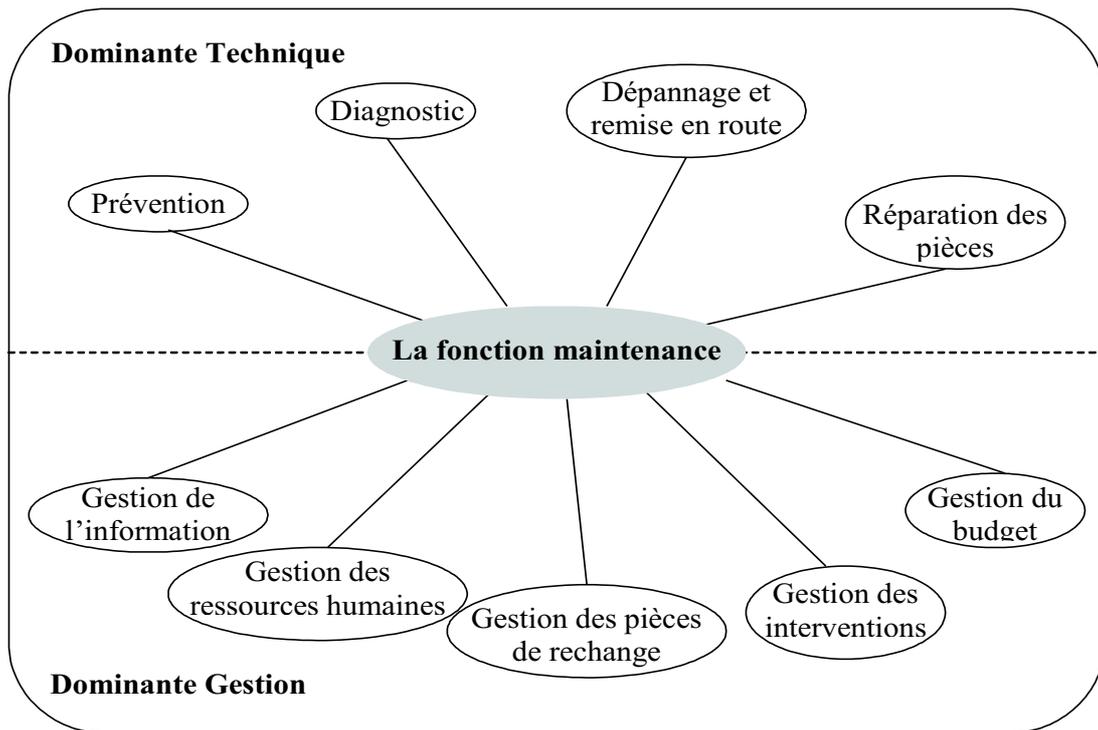


Figure 1.2 – Principaux domaines d'action de la fonction maintenance

sont liés. Comme l'a relevé Salith O. Duffuaa, éditeur de la revue *Journal of Quality in Maintenance Engineering*, il apparaît notamment les problèmes suivants :

- organisation de la maintenance et gestion du personnel,
- qualité et stratégies de maintenance,
- maintenance préventive et modèles de remplacement,
- maintenance prédictive,
- budgétisation et contrôle de coût,
- modélisation et optimisation des systèmes de maintenance,
- planification et ordonnancement de la maintenance,
- planification des charge (capacité) de maintenance,
- approvisionnement et gestion de stocks en maintenance,
- maintenance basée sur la fiabilité,
- contrôle qualité, processus de capabilité et maintenance,
- maintenance productive totale,
- techniques de maintenance conditionnelle,
- systèmes d'information en maintenance,
- gestion du coût d'exploitation des systèmes,
- analyse des performance et standards en maintenance,
- maintenance et nouvelles technologies

pour ne citer que ceux-là. Ces différents problèmes paraissent à divers niveaux de décision et demandent des compétences aussi variées que le sont les natures des problèmes.

1.3.2 Les différentes formes de maintenance

Les différentes pratiques de maintenance sont groupées en deux catégories principales que sont la maintenance à la suite d'une défaillance (la maintenance curative principalement) et la maintenance entrant dans le cadre de la prévention des défaillances (la maintenance préventive systématique et la maintenance conditionnelle, et il y a quelques années, la maintenance prédictive). Les différentes formes de maintenance peuvent être synthétisées comme sur le graphe 1.3 [111]. Cette figure, très explicite, représente de façon concise et précise les grands groupes de maintenance en pratique. Il faut noter que les actions de maintenance corrective ne peuvent être entièrement éliminées dans les systèmes de productions mais une bonne politique de maintenance préventive permet de réduire considérablement la nécessité des actions de maintenance corrective. D'après Monchy [110], les objectifs de la maintenance préventive peuvent être résumés comme suit.

1.3.3 Objectifs visés par le « préventif »

Les politiques de maintenance préventive visent des buts spécifiques parmi lesquels :

- augmenter la fiabilité d'un équipement, donc réduire les défaillances en service : réduction des coûts de défaillance, amélioration de la disponibilité.
- augmenter la durée de vie efficace d'un équipement.
- améliorer l'ordonnancement des travaux, donc les relations avec la production.
- réduire et régulariser la charge de travail.
- faciliter la gestion des stocks (consommations prévues).
- assurer la sécurité (moins d'improvisations dangereuses).
- plus globalement, en réduisant la part de « fortuit », améliorer le climat des relations humaines (étant donné qu'une panne imprévue est toujours génératrice de tensions).

Certes, dès lors que l'on connaît précisément le comportement d'un système, une bonne planification de la maintenance préventive (systématique) peut être faite efficacement. Mais la maintenance conditionnelle est souvent utilisée en cas de comportement suspects des équipements. Une bonne politique de maintenance conditionnelle se base sur une bonne politique d'inspection.

1.3.4 Avantages et limites de la maintenance conditionnelle

Une bonne politique de maintenance conditionnelle permet [33] :

- une augmentation de la longévité du matériel,
- un contrôle du matériel mieux géré,

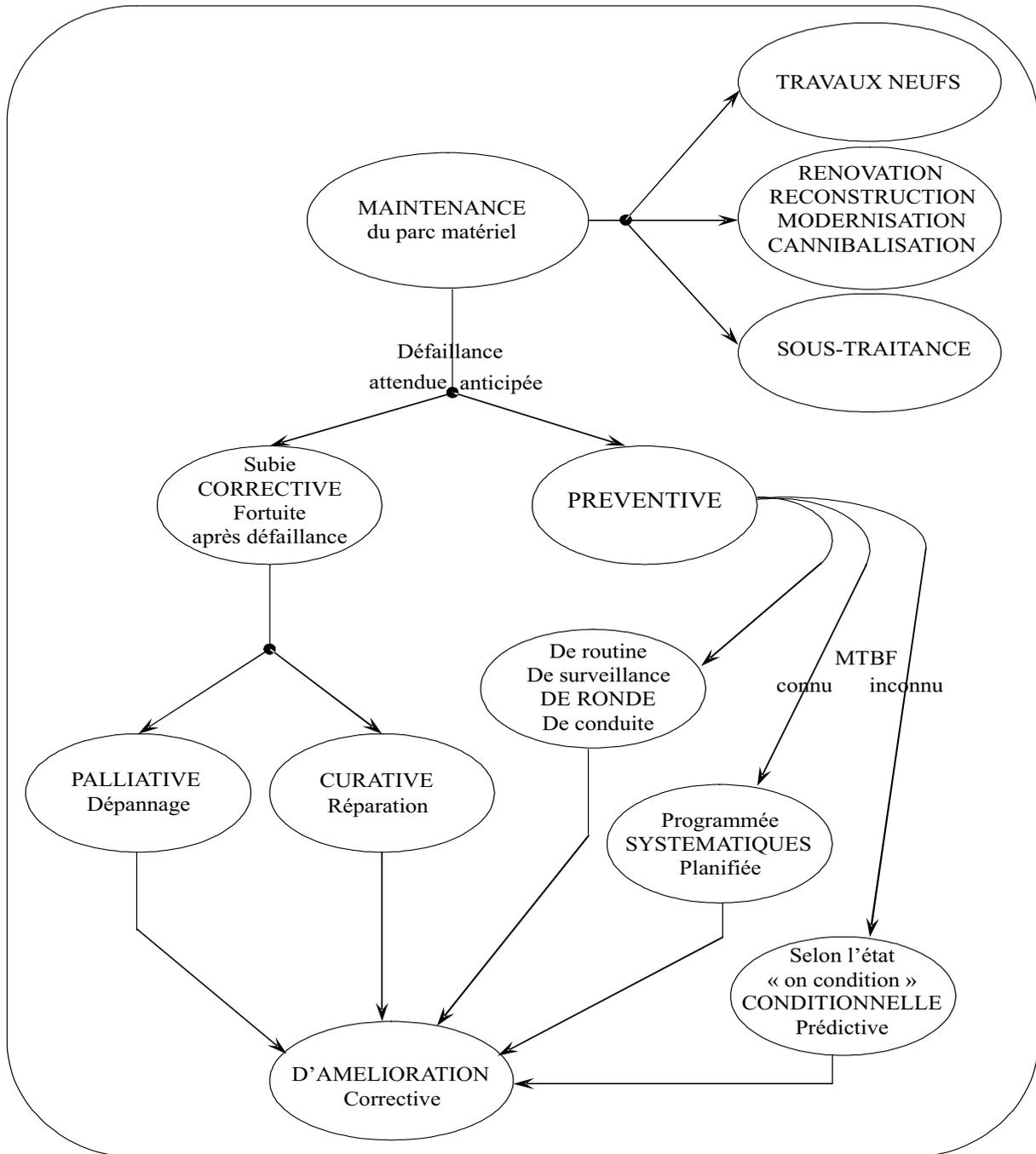


Figure 1.3 – Graphe des différentes formes de maintenance

- un coût de réparation moins élevé,
- une amélioration de la productivité de l'entreprise,
- une diminution des stocks de production,
- la limitation des pièces de rechange,
- une amélioration de la sécurité,
- une amélioration de la crédibilité du services d'entretien,
- une plus grande motivation de personnel maintenance,
- une image de marque de la société rehaussée,

- de meilleures conditions d'assurance.

Certaines limites sont à détecter pour la maintenance conditionnelle, principalement sur le plan technique [33]. L'efficacité d'une maintenance conditionnelle est subordonnée à l'efficacité et à la fiabilité des paramètres de mesure qui la caractérisent. La maintenance conditionnelle sera donc réservée aux matériels dont l'évolution d'éventuels défauts est facilement détectable et mesurable avec des capteurs fiables. Il s'agira principalement :

- des machines en mouvement (capteurs vibrations-analyse huiles ...)
- des canalisations (détection de fuites ...)
- des équipements soumis à température ou pression (capteurs)
- des capteurs électriques (analyse chromato-graphique ...)
- des appareils soumis à déformations (examen de structure ou de surface).

Outre les pratiques traditionnelles de la maintenance, il existe des pratiques particulières de maintenance. Compte tenu des spécificités des entreprises et des objectifs généraux envisagés, différentes pratiques de maintenance ont été développées et appliquées dans les entreprises. Les plus répandues et les plus utilisées sont la maintenance basée sur la fiabilité (RCM - Reliability-Centred Maintenance) développée aux États-Unis et la maintenance productive totale (TPM - Total Productive Maintenance) développée par les Japonais.

1.3.5 La maintenance basée sur la fiabilité ou RCM

La maintenance basée sur la fiabilité (RMC - Reliability Centred Maintenance) est une technique de maintenance qui a fait son apparition dans l'industrie aéronautique vers la fin de années 1960 et au début des années 1970 aux États-Unis (voir dans [113], [120], [129], [143], [154], [175]). Cet outil a le risque à l'œil en considérant de manière fort structurée, les conséquences et la probabilité d'une défaillance. La RMC essaie de minimiser les conséquences d'une défaillance en cherchant et en exécutant les tâches critiques de maintenance. En particulier, elle identifie des défaillances cachées et aide à améliorer la disponibilité.

L'objectif principal de la maintenance basée sur la fiabilité est de réduire le coût de la maintenance en se focalisant essentiellement sur les fonctions les plus importantes du système, tout en reléguant au second plan ou en évitant les actions de maintenance qui ne sont pas strictement nécessaires.

La maintenance basée sur la fiabilité est une approche de maintenance préventive. Elle est basée sur l'hypothèse que la fiabilité inhérente à un équipement est fonction la conception et de la qualité de construction [128]. Une maintenance préventive efficace devra assurer que cette fiabilité soit réalisée. Une analyse de RCM à la base permet de trouver des réponses aux questions ci-après.

1. Quelles sont les fonctions et les performances standards associées de l'équipement dans son contexte opérationnel présent ?
2. Dans quels cas faillit-il à assurer ces fonctions ?

3. Quel est la cause de chaque défaillance fonctionnelle ?
4. Que se passe-t-il lorsque survient chaque défaillance ?
5. Quel est le degré de gravité de chaque défaillance ?
6. Que peut-on faire pour éviter chaque défaillance ?
7. Que devra-t-on faire lorsqu'une intervention de maintenance préventive adéquate ne peut-être réalisée ?

D'après Rausand [128], et Moudray [113], une analyse de RCM peut être réalisée comme une séquence d'activités dont notamment :

1. la préparation de l'étude,
2. la sélection et définition du système,
3. l'analyse de défaillance fonctionnelle (FFA - Functional Failure Analysis),
4. la sélection de l'entité critique,
5. la collecte et l'analyse des données,
6. l'analyse des modes de défaillance, de leurs effets et de leurs criticités (AMDEC) (FMECA - Failure Modes, Effects and Criticality Analysis),
7. la sélection des actions de maintenance,
8. la détermination des intervalles de maintenance,
9. l'analyse de comparaison de la maintenance préventive,
10. le traitement des entités non critiques,
11. l'implémentation,
12. la collecte et la remise à jour des données en phase opérationnelle.

Outre la maintenance basée sur la fiabilité, une autre pratique de maintenance largement utilisée est la maintenance productive total (TPM).

1.3.6 La maintenance productive totale ou TPM

Initiée par les Japonais et mise en œuvre à partir des années 1970, la maintenance productive totale (TPM - Total Productive Maintenance) est une politique de maintenance qui se répands de plus en plus. Elle est en grande partie due à l'automatisation croissante des systèmes d'exploitation. Elle a pour but d'impliquer tout le personnel de l'entreprise (des plus hauts cadres jusqu'aux ouvriers) dans l'action de « maintenance » (voir dans [118], [119], [147], [150]). Elle comprend des stratégies visant notamment à :

- maximiser l'efficacité (rendement) globale de l'entreprise,
- établir un système compréhensif de maintenance préventive qui couvre le cycle de vie global de l'équipement,
- impliquer la participation de toutes les divisions dans les activités de maintenance,
- impliquer la participation effective de tous le personnel, de la direction de l'entreprise à tous les services d'exploitation,

- promouvoir la maintenance préventive par la motivation des cercles d'activité.

Le mot « Total » dans la dénomination vient non seulement de l'objectif de rendement global (total) envisagé mais aussi de l'implication de tout le personnel dans l'action de maintenance. La maintenance productive totale recherche donc, avec un minimum de coût global d'exploitation (LCC - Life Cycle Cost) de l'équipement, à retirer le meilleur résultat de celui-ci. Pour cela, elle doit affronter les différentes sources de pertes qui nuisent au rendement optimum à savoir :

- les pertes dues aux arrêts (la panne, le réglage avant démarrage),
- les pertes par ralentissement (la marche à vide, les petits arrêts, le ralentissement),
- les pertes dues aux défauts (les défauts internes au processus,...).

La mise en place d'une politique de TPM se fait généralement en quatre phases qui se subdivisent en 12 étapes. Ces phases sont :

1. la phase de préparation,
2. la phase de mise en application,
3. la phase d'exécution de la politique,
4. la phase de fixation et de perfectionnement de la pratique.

Étant donné l'implication de toute l'entreprise, chacune de ces phases est nécessaire afin de faire entrer progressivement la pratique du TPM dans les habitudes de tout le personnel.

1. La phase de préparation est subdivisée en cinq étapes qui sont :

- déclaration de la décision prise par les dirigeants d'appliquer la TPM à tous les acteurs impliqués,
- formation à la TPM et campagne de promotion à tous les échelons concernés,
- organisation d'une structure adaptée à la TPM,
- définition des orientations et des principes de la TPM et des objectifs visés,
- établissement du projet directeur pour l'application de la TPM.

2. La phase de mise en application est la sixième étape de la mise en application de la TPM. C'est le « kick-off » (coup d'envoi) de la TPM. On s'attaque alors à l'élimination de toutes les sources de perte énumérées plus haut.

3. La phase d'exécution de la politique se subdivise en cinq étapes telles que :

- Promotion des améliorations pour le meilleur rendement ;
- Organisation de l'auto-maintenance par les exploitants ;
- Organisation de la maintenance programmée de la division maintenance ;
- Formation technique d'exploitation et de maintenance ;
- Organisation pour la gestion de données de l'installation pendant la période initiale.

4. La phase de fixation et de perfectionnement de la pratique. Elle consiste en l'application complète de la TPM et de l'amélioration du niveau de pratique.

Cette pratique de motivation du personnel permet une meilleure exploitation des équipements avec des gestes simples (lubrification, nettoyage, surveillance d'apparition d'anomalies,...) qui permettent d'éviter au mieux un fonctionnement critique.

Dans toutes ces pratiques de la maintenance, il y a une fonction qui est présente à tous les niveaux de décision dans le processus de maintenance. C'est la fonction ordonnancement.

1.3.7 La fonction ordonnancement en maintenance

La mise en œuvre d'une politique de maintenance va de la planification, à l'ordonnancement, puis à l'exécution des travaux de maintenance. Comme l'a relevé Francastel [65], la fonction ordonnancement a pour but d'organiser (ordonnancer) les travaux de maintenance et de les lancer avec les moyens disponibles adaptés à un optimum technico-économique. L'ordonnancement des travaux est l'ensemble des dispositions méthodiques à prendre pour choisir, organiser, gérer, optimiser... les moyens en personnels et en matériels les mieux adaptés aux interventions requises [47], [65], [66], [91], [92], [110], [111]. Les travaux de maintenance, préventivement définis ou nécessités par des opérations fortuites, doivent atteindre un optimum, visant à la fois la qualité des interventions requises, la sécurité des interventions, les délais requis, le moindre coût. La fonction ordonnancement procure à la fin les éléments de retour d'expériences.

D'après Monchy [111], l'organisation de la maintenance, en général et l'ordonnancement en maintenance, en particulier, fait face à des difficultés particulières à cause de certaines spécificités de cette organisation. Il est incontestable que la diversité des actions de maintenance complique leur programmation :

- d'une durée de quelques secondes en auto-maintenance à des milliers d'heures pour les chantiers lourds de type arrêt périodique d'unité de production. Dans le premier cas, le besoin du client est satisfait par une grande réactivité. Dans le second, il est satisfait par le respect du délai et la qualité du travail ;
- du systématique planifié six (6) mois à l'avance au correctif fortuit, au mauvais endroit au mauvais moment ;
- de l'urgent à faire sans délai à l'amélioration dépourvue de délai et toujours remise à demain.

L'organisation à mettre en œuvre doit tenir compte de cette variabilité : légère et souple pour assurer la réactivité des équipes de terrain, logistique lourde pour la gestion des arrêts. Une autre difficulté est que la production étant maîtresse et ses arrêts planifiés, la maintenance se retrouve logiquement dans une situation de dépendance. Une bonne synchronisation est indispensable entre les services afin de profiter au mieux des arrêts de production pour réaliser des opérations de maintenance. En outre, l'éclatement des installations sur différents sites souvent très dispersés sur le plan géographique caractérise de plus en plus les entreprises. Cette dispersion est une autre source de difficultés pour l'organisation de la maintenance.

Ainsi, dans un service de maintenance, la fonction ordonnancement est particulièrement lourde

et délicate. Les travaux sont variés et la notion d'urgence est plus fréquente et plus lourde de conséquence qu'en production. La principale spécificité de l'ordonnancement des travaux de maintenance réside dans le caractère *fortuit* d'un flux d'interventions, souvent de dépannages et souvent d'urgence. L'ordonnancement de la maintenance a donc des problèmes spécifiques à résoudre notamment :

- La dépendance de la production (arrêts de fabrication) ;
- La prise en compte de la sécurité ;
- Le suivi des travaux sous-traités, souvent nombreux en maintenance ;
- L'approvisionnement des fournitures et pièces de rechange ;
- Les moyens de manutention spéciaux à rendre disponibles ;
- Le tri des urgences d'interventions surtout correctives.

Sur le plan du planning de charge des travaux de maintenance, les objectifs à atteindre sont :

- La définition de la meilleure adéquation « moyen-besoin ».
- L'optimisation de l'effectif par catégorie professionnelle ou par équipe.
- La prévision des besoins en sous-traitance ou en renfort temporaires.
- La gestion et le respect des délais des programmes.
- Le suivi de l'avancement global des programmes.

L'approche la plus utilisée en ordonnancement de la maintenance actuellement est la représentation par diagrammes de Gantt pour la représentation de quelques activités ponctuelles. Pour des activités un peu plus lourdes, où la représentation par diagrammes de Gantt n'est plus possible, la méthode PERT (Program Evaluation and Review Technique) est utilisée [111]. C'est une méthode basée sur la théorie des graphes, mise au point aux États-Unis en 1958 pour l'ordonnancement du programme spatial « Polaris ». Elle comporte plusieurs variantes et est particulièrement facile à traiter par ordinateur. Elle peut être efficace pour l'ordonnancement des activités de maintenance dans un atelier. Mais pour des systèmes complexes (comme c'est le cas par exemple pour les systèmes distribués), elle devient inappropriée. Dans ces cas, des outils adaptés doivent être développés. D'autres outils de GMAO (Gestion de Maintenance Assistée par Ordinateur) permettent surtout une bonne tenue des bases de données nécessaires pour la conduite de la fonction maintenance dans l'entreprise.

1.4 Conclusion

Comme on peut le voir dans ce chapitre, la maintenance a beaucoup évolué en tant que méthode de maîtrise du coût de vie des biens humains consommables lors de la deuxième moitié du vingtième siècle. Cette évolution peut se voir dans sa conception, dans sa formalisation, dans sa mise en œuvre et dans les diverses pratiques qui se sont développées au cours de ce demi-siècle. Cette évolution rapide de la pratique de la maintenance s'est faite en même temps que la recherche scientifique développe des approches afin de trouver des solutions aux différents problèmes qui

ont jalonné et jalonnent encore la pratique de la maintenance, notamment dans les industries manufacturières. Le chapitre suivant sera justement consacré à ces développements scientifiques de la maintenance dans la recherche.

Ce document est globalement structuré comme suit. Après ce premier chapitre d'introduction à la maintenance, nous analysons dans le second chapitre l'état de l'art de la maintenance dans la recherche scientifique. Après cette analyse de l'état de l'art, nous tirerons les conclusions de l'existant puis présenterons dans le même chapitre 2 la problématique de nos travaux, les raisons nécessitant ces travaux et les objectifs envisagés. Ce positionnement permettra de mieux cerner la suite de ce document. D'ores et déjà, il faut noter que le chapitre 3 sera consacré à des propositions d'approches de prise de décision en temps-réel dans un contexte de système distribué dont l'objectif est de minimiser les coûts dus à des états critiques des entités du système considéré. Ensuite, nous aborderons dans le chapitre 4 des problèmes de prise de décision de maintenance en temps-réel toujours dans le contexte de système distribué, avec des contraintes supplémentaires de délais sur les interventions de maintenance. Ces coûts sont dus notamment, d'une part, à des combinaisons des états critiques des entités du système et des retards des tâches de maintenance, et, d'autre part, à des coûts des fréquences élevées ou de déficit de maintenance. Dans ces deux chapitres, il est supposé que la maintenance est évaluée sur un horizon fini et que sur cet horizon, les tâches de maintenance sont entreprises au besoin et dès que possible. Le problème pourra être abordé sous une forme différente où les tâches de maintenance ne peuvent pas être entreprises à tout moment sur l'horizon considéré, mais sont plutôt confinées dans des fenêtres temporelles à l'extérieur desquelles la production est prioritaire. Les approches proposées sont reconsidérées dans le chapitre 5 dans une étude de robustesse. Le chapitre 6 vient apporter une conclusion générale sur les travaux présentés dans ce document et présente quelques perspectives de recherche. Nous présentons dans l'appendice A quelques notions de base utilisées dans ce document. Ensuite, afin d'envisager l'utilisation de la fonction disponibilité des entités dans des cas autres que celui de la loi exponentielle souvent utilisées dans la littérature, une analyse exhaustive de cette fonction disponibilité dans des cas non exponentiels est réalisée dans l'appendice B.

Chapitre 2

La maintenance dans la recherche scientifique : état de l'art

2.1 Introduction

Après le chapitre précédent consacré à l'évolution de la maintenance depuis ses débuts et aux différentes pratiques de maintenance dans les activités humaines, nous nous intéressons dans le présent chapitre à l'évolution de la maintenance dans la recherche scientifique. Comme nous l'avons relevé dans le chapitre 1, l'évolution de la maintenance a suscité un grand intérêt dans la recherche scientifique. La modélisation et le développement d'approches pour la prise de décision en maintenance sont devenus des domaines importants de la recherche. L'industrie a clairement reconnu l'importance de la maintenance mais a été lente à adopter des approches scientifiques. La raison de cet état de chose est sans doute liée au fait que la maintenance a été longtemps regardée comme une activité marginale dans beaucoup de domaines ; sa valeur se retrouve dans des résultats à moyen et à long terme, contrairement à la recherche du bénéfice immédiat qui est l'objectif principal de la production. Les travaux sur la maintenance sont groupés dans diverses catégories que nous tenterons d'analyser dans ce chapitre.

L'optimisation de la maintenance consiste globalement en des modèles mathématiques servant à trouver soit le meilleur compromis possible entre le profit de la maintenance et son coût ou les moments idéaux pour exécuter les interventions de maintenance. Les différents états de l'art que nous avons mentionnés dans le chapitre 1 et sur lesquels nous reviendront, révèlent abondamment les travaux sur chacun de ces aspects de l'optimisation de la maintenance. Ces différents états de l'art révèlent que l'optimisation de la maintenance est un domaine scientifique bien établi, domaine auquel aussi bien les ingénieurs que les scientifiques ont largement contribué.

Nous allons, dans ce chapitre, donner un aperçu de la situation de la maintenance dans la recherche scientifique. Les travaux sont très abondants et se regroupent suivant différents points de vue. Nous les regroupons suivant les aspects de maintenance abordés, les objectifs visés, les approches de résolution proposées. Dans chaque groupe de travaux, nous aborderons les travaux les plus représentatifs et nous donnerons un aperçu du reste dans la conclusion de ce chapitre. Dans

le reste de ce chapitre, nous allons, dans la section suivante, parler des classifications des travaux proposés par plusieurs études de l'état de l'art. Puis, nous parlerons des pratiques de maintenance conditionnelle basée notamment sur les inspections et les remplacements. Nous enchaînerons avec les travaux de la littérature qui utilisent le concept de réparation minimale avant d'aborder les travaux qui étudient le regroupement des tâches de maintenance en classes puis le problème spécifique dans la maintenance consistant à allouer des ressources pour les opérations de maintenance. Cela nous amènera à une catégorie particulière d'approches de gestion des tâches de maintenance en se basant notamment sur des méthodes de gestion de production. Nous terminerons ce chapitre par la conclusion dans laquelle nous tirerons les principales observations de l'état de l'art avant de positionner nos travaux.

2.2 Critères de classification des problèmes de maintenance

Les problèmes de décision en maintenance peuvent être classifiés suivant plusieurs aspects. Différents aspects de classification peuvent être observés dans les nombreuses études sur l'état de l'art des travaux sur la maintenance telles qu'on peut le voir dans la littérature [38], [49], [57], [56], [52], [73], [86], [107], [109], [125], [126], [135], [140], [141], [151], [156], [165] et [168] que nous avons énumérés précédemment.

Comme l'ont souligné Dekker et Scarf [52], une classification courante est celle relative à l'horizon de temps considéré. Selon eux, les considérations de maintenance devront commencer dès la phase de conception des systèmes. Le type d'équipement, son niveau de redondance ou son accessibilité affectent alors considérablement la maintenabilité. Dès l'acquisition des systèmes, les coûts futurs de maintenance devront aussi être pris en compte. Pour couvrir les coûts dans toutes les phases d'exploitation du système, le concept de « coût global du cycle de vie » ou « coût global d'exploitation » (life cycle costing) a été introduit. Le concept ou la stratégie de maintenance décrit quel événement (défaillance, durée d'exploitation) engendre quel type de maintenance (inspection, réparation, remplacement). Cette stratégie de maintenance peut être déterminée aussi bien dès la phase de conception que dans la phase d'exploitation du système. La plupart des modèles mathématiques se concentrent sur cet aspect du problème, surtout sur la détermination de l'intervalle de temps (périodes) pour les interventions d'inspection ou de maintenance préventive. Une fois qu'un système est en phase opérationnelle, la maintenance doit être planifiée et les tâches de maintenance ordonnancées. La planification des activités de maintenance devra être faite en concordance avec les autres activités (en l'occurrence la production). Une fois la planification de la maintenance faite, l'ordonnement intervient pour déterminer l'ordre d'exécution des différentes tâches. L'ordonnement inclus la détermination des priorités et l'utilisation des ressources disponibles le plus efficacement possible. Il y a enfin le contrôle de la maintenance, qui consiste à comparer les retombées de la maintenance avec les plans établis tout en indiquant aux managers là où il y a des problèmes ; cet aspect porte les indicateurs de performance du système.

Un autre critère de classification des travaux de maintenance est le niveau où la décision doit être prise. On peut distinguer les niveaux suivants : au niveau du groupe industriel ou de la compagnie,

au niveau de l'usine, au niveau du système, au niveau du composant. Au niveau du groupe ou de la compagnie, les décisions doivent être prises compte tenu de la taille de la compagnie. C'est le cas notamment dans le génie civil avec les routes, ponts, barrages, etc., où le budget vient, sinon du gouvernement, du moins de son représentant local. Cette décision intervient alors à chaque niveau. La conséquence de cette classification est que le niveau le plus élevé doit prendre en compte les besoins de tous les niveaux les plus bas. Au niveau de chaque système, une évaluation indépendante de la valeur économique peut être faite et les besoins budgétaires de la maintenance déterminés. C'est cette évaluation qui est donc prise en compte aussi bien au niveau le plus haut qu'au niveau le moins élevé.

Les différents aperçus de l'état de l'art énumérés ci-dessus peuvent être classés suivant le (ou les) critère(s) de classification utilisé(s) par les auteurs pour regrouper les travaux. Certains de ces aperçus de la littérature analyse les modèles suivant leurs applications éventuelles. C'est le cas des papiers de Dekker [49], de Scarf [135], de Dekker et Scarf [52]. D'autres, par contre, font des analyses suivant l'approche méthodologique suivie. Dans ce cadre, Thomas [151], et Wang [165] concentrent leurs analyses sur les politiques de remplacement en vue de la maintenance conditionnelle, alors que Wijnmalen et Hontelez [168] mettent plutôt l'accent sur les travaux proposant des décisions sur des approches markoviennes. Jardine et Buzacott [86] eux regardent les pratiques de recherche opérationnelle pouvant permettre l'amélioration de la fiabilité des systèmes par des politiques adéquates de maintenance. Au delà de la recherche opérationnelle, Sherwin [141], d'une part, et Pintelon et Gelders [126], d'autre part, analysent les outils de management en général et d'aide à la décision en particulier dans les processus de maintenance. Quant à Cho [38], il met l'accent sur les modèles de remplacement pour les systèmes de plus d'un composant.

Nous allons maintenant aborder les différents aspects dominants des travaux de la maintenance dans la littérature. Quoique d'autres points de vue puissent exister, nous retenons ceux qui ont fait le plus l'objet d'attention dans la littérature. L'un des domaines de la maintenance ayant retenu le plus l'attention dans la recherche scientifique est la maintenance conditionnelle qui est l'objet de la section suivante.

2.3 Politiques d'inspection et maintenance conditionnelle

Des données fiables de la maintenance, qui sont des entrées nécessaires à la mise en application des modèles, sont en général difficiles à obtenir et leur collecte est souvent incomplète. De ce fait, les décisions de maintenance peuvent être basées sur des observations de comportements suspects des systèmes. La maintenance conditionnelle basée sur l'observation de ces comportements suspects, est un aspect de la maintenance largement discutée dans la littérature. Afin de mettre en œuvre une politique de maintenance conditionnelle efficace et dans le souci d'anticiper au mieux les défaillances, il est nécessaire d'observer les paramètres sur lesquels se base cette politique avant et lors des défaillances. Cependant, les défaillances sont rares et les seules observations basées sur ces défaillances peuvent être insuffisantes. Pour cela, des politiques d'inspections sont pratiquées

pour déterminer ces anomalies. Une bonne politique de maintenance conditionnelle est donc intrinsèquement liée à une bonne politique d'inspection. Les travaux sur la maintenance conditionnelle sont nombreux dans la littérature. Nous citerons à titre d'exemple la thèse de Castanier [34].

2.3.1 Politiques d'inspection

En général, les niveaux de dégradation des systèmes ne peuvent être observés continuellement dans le temps. Les politiques d'inspection permettent de faire des analyses importantes sur l'état des systèmes en vue des interventions de maintenance appropriées allant des interventions de maintenance préventive à des remplacements.

C'est dans cette optique que Shahani et Newbold [139] proposent une approche de politique d'inspection pour des systèmes dont la fiabilité suit la loi de Weibull. Ils reposent leur approche sur la caractérisation d'une probabilité de défaillance entre deux inspections consécutives et cherchent à minimiser le coût global jusqu'à la détection d'une panne. Tadikamalla [148] lui base son modèle sur des systèmes dont la durée de vie suit une loi Gamma. Les travaux de Menipaz [108] combinent des inspections séquentielles et des inspections continues suivant l'âge du système. Leur approche pour résoudre les problèmes posés se basent notamment sur la méthode de programmation dynamique. Dans ce même sens, Kaio et Osaki [87] proposent des comparaisons entre différentes politiques d'inspection et proposent des modifications afin d'améliorer certaines caractéristiques de ces politiques. Munford [114] fait aussi des comparaisons et propose des politiques d'inspection. Outre le modèle de Shahani et Newbold, il préconise des inspections périodiques et des inspections avec des durées inter-inspection proportionnelles à la durée de vie résiduelle moyenne des systèmes. Quant à Luss [104], [105] [106], il propose une généralisation des approches de Munford en utilisant la programmation dynamique pour déterminer des séquences d'inspection. Dans cette catégorie d'inspection périodiques pour la maintenance conditionnelle, notons aussi les travaux de Barbera *et al.* [20] reposant sur un modèle exponentiel suivant des intervalles fixés et aussi les travaux de Nakagawa [117] pour des inspections périodiques en vue de la maintenance préventive.

Une intervention de maintenance sur un système peut occasionner des arrêts du système et donc des pertes économiques dues à ces activités. Il est donc nécessaire que les durées des interventions de maintenance soient prises en compte dans les approches proposées pour résoudre les problèmes de maintenance en général et d'inspection en particulier. Mais dans la littérature très peu de travaux considèrent cet aspect dans les modèles d'inspection. Luss [104] évalue à l'aide de résultats de la théorie de renouvellement plusieurs critères moyens asymptotiques pour lesquels les durées d'intervention restent constantes. Il retrouve, avec des méthodes basées notamment sur la programmation dynamique à partir de l'espérance du coût cumulé entre deux inspections, les propriétés des intervalles inter-inspection optimaux. Ces propriétés sont fonctions de l'âge du système et de son taux de défaillance. En général, les méthodes basées sur les processus de renouvellement composés prennent en compte des durées aléatoires d'intervention entre une panne et la remise en service suivante du système mais ils traitent souvent des systèmes à un seul composant.

D'autres travaux sur les politiques d'inspection font des couplages entre politiques d'inspection

et politiques de remplacement. C'est ce que nous abordons dans la sous-section ci-après.

2.3.2 Remplacement et politique de maintenance conditionnelle

Les politiques d'inspections peuvent conduire à des remplacements, même si l'état du système n'est pas critique. On parle alors de remplacement préventif. Les règles de remplacement préventif peuvent être définies selon le niveau de fonctionnement du système ; si ce niveau se trouve dans une zone fixée a priori, alors le système est remplacé. La règle de remplacement préventif est généralement liée à un seuil critique de performance du système.

Christer et Wang [40] ont construit un modèle d'inspection permettant un remplacement préventif dès que le niveau de dégradation observé lors d'une inspection est compris entre un seuil donné et la limite de panne. D'autres modèles de remplacement préventif sont basés sur l'âge du système inspecté. La détermination des instants d'inspection repose alors sur la minimisation de l'espérance du coût total par unité de temps sur un intervalle de temps défini par l'inspection courante et la date de la prochaine inspection. D'autres modèles de remplacement préventif sont proposés par Christer [39], Turco et Parolini [155], Okumura *et al.* [122] notamment.

La maintenance d'un système est en général réalisée en fonction de paramètres indépendants de l'évolution propre du système (indisponibilité des ressources de maintenance, etc.). Elle doit donc se soumettre à ces paramètres desquels dépend aussi la performance de la politique menée. Dekker et Dijkstra [50] restreignent la possibilité de remplacement préventif uniquement aux dates exponentiellement réparties. De ce fait, elles ne dépendent pas seulement des paramètres propres du système mais aussi de paramètres exogènes. Iskandar et Sandoh [85] eux proposent une politique de remplacement basée sur l'opportunité. L'analyse du coût marginal est aussi un élément beaucoup utilisé pour la caractérisation des politiques de remplacement. Elle repose sur la détermination de la fonction coût marginal qui est l'espérance du coût supplémentaire engendré si on repousse l'intervention de maintenance d'un différentiel de temps par rapport à la date initialement prévue. Cette fonction influence directement le coût moyen de maintenance à long terme. C'est l'objet par exemple de travaux de Aven et Dekker [16], de Frenk *et al.* [67], de Berg [23]. Ces derniers proposent notamment des conditions nécessaires à l'existence d'une politique optimale dans une classe de politiques de remplacement déterminées sur la base de l'âge. D'autres modèles de politiques de remplacement existent et se basent sur différents processus stochastiques. Il y a notamment des modèles markoviens et semi-markoviens comme dans les travaux de Feldman [62], Kao [88], de Kolesar [94], Sielersleben [142]. Il y a aussi des modèles de remplacement par bloc. Notons pour cela les travaux de Archibald et Dekker [13], Kolesar [94], Lam et Yeh [98], [99], [97], Lam [100]. Et nous n'oublions pas les modèles basés sur les processus de renouvellement pour les politiques de remplacement, la théorie de comptage utilisée notamment par Aven [15]. Il faut souligner l'introduction de durées déterministes dans les politiques de remplacement par exemple dans les travaux de Zuckerman [171], [173], [172].

Parmi les travaux sur les politiques de remplacement, nous tenons à faire remarquer la particularité de certains. Cette particularité consiste à combiner des anciens et de nouveaux composants

comme on peut le voir dans les travaux de Lam [100], et de Stadge et Zuckerman [146]. En général, ces travaux utilisent des processus de renouvellement à délai. C'est le cas de Lam dont l'approche considère un processus de renouvellement à délais pour la planification de la maintenance d'un système qui au départ, n'est pas nécessairement neuf ; ce système doit être remplacé par un système neuf après un temps de fonctionnement. Son approche permet de modéliser les instants de régénérescence d'un tel système.

Dans la section suivante, il y a d'autres travaux sur les politiques d'inspection, de remplacement en vue de la maintenance conditionnelle. Ces travaux tout comme certains travaux sur les politiques de réparations préventives et correctives, se basent sur des informations imparfaites, des réparations partielles, ou encore de la maintenance imparfaite. Nous les regroupons dans le cadre de maintenance imparfaite dans cette section.

2.4 Réparation partielle et politiques de maintenance imparfaite

Dans la section précédent, la politique de maintenance consiste à prendre une des deux décisions suivantes à la suite d'une inspection : remplacer le système inspecté ou le laisser continuer son fonctionnement dans le même état. C'est en quelque sorte une politique du "tout ou rien". Le coût d'un remplacement est souvent élevé et une solution intermédiaire consiste à réparer le système ou son composant en panne. En général, elle est fait l'hypothèse de remplacement parfait dans les travaux. Contrairement à cela, on a les réparations minimales.

Le concept de réparation minimale stipule que lorsqu'un système tombe en panne, il est immédiatement restauré dans l'état où il était juste avant la défaillance, ses caractéristiques étant alors remises dans les mêmes conditions qu'avant la défaillance, souvent afin de pouvoir aller jusqu'au bout d'une fonction entamée. C'est ce qu'on appelle la maintenance palliative en pratique. Les réparations minimales permettent de remettre un système dans un état de bon fonctionnement qui est généralement observé juste avant la défaillance. À l'opposé de la réparation minimale, les politiques de remplacement en général suggèrent de remplacer entièrement un système ou un sous-système en panne. Dans beaucoup de situations, le remplacement implique un coût nettement plus important. La question qui se pose alors est celle de la date optimale de remplacement. C'est à cette question que les différents travaux énumérés dans les sections précédentes ont tenter de répondre suivant différentes situations.

Différents travaux de la littérature ont fait usage de ce concept de réparation minimale. Tahara et Nishida [149] ont proposé un modèle de coût de défaillance dans lequel ils caractérisent les remplacements préventifs et les remplacements à la suite d'une défaillance en introduisant un coût additionnel pour la panne. Ils considèrent un système fonctionnant continuellement et définissent une politique de type (t, T) ; c'est une politique dans laquelle la réparation minimale est utilisée pour toute défaillance avant la date t , un remplacement pour toute défaillance entre les dates t et T , et si le système est toujours en fonctionnement à la date T , il est systématiquement remplacé à cette date. Ils démontrent une politique (t, T) optimale pour le modèle de coût considéré et proposent une politique sur le critère de coût moyen à long terme. Quant à Tilquin et Cléroux [152], ils proposent

un modèle de coût d'ajustement qui peut inclure des coûts d'ajustement périodiques, des coûts de dépréciation ou autre, alors que Boland et Proschan [28] analysent un modèle similaire lorsque le coût de réparation minimale est une fonction croissante du nombre de réparations minimales du système depuis le dernier remplacement du système. Boland [27] propose une autre variante dans laquelle le coût de réparation minimale dépend de l'âge de la machine plutôt que du nombre de réparations minimales. Quant à Bagai et Jain [18], ils étudient un modèle de remplacement optimal pour un système soumis à des réparations minimales.

Afin d'apporter quelques modifications par rapport aux modèles standards de réparations minimales existants, Chen et Feldman [37] proposent un modèle dans lequel, en plus des coûts opératoires dépendant de l'âge, la réparation minimale peut échouer. Ils imposent aussi la limitation que la réparation minimale ne doit être utilisée qu'une seule fois. En revanche, ils supposent que les durées des réparations minimales sont négligeables. La politique qu'ils ont développée est similaire à celle de Tahara et Nishida. Pour leur part, Ohnishi *et al.* [149] eux développent un modèle de réparation minimale pour un système partiellement observable alors que Park [124] développe une approche pour optimiser le nombre minimale de réparations minimales avant que ne survienne un remplacement, de même que Segawa *et al.* [136].

En dehors de ces travaux, d'autres modèles de réparations minimales peuvent être trouvés dans les travaux de Block *et al.* [25, 26], de Cléroux *et al.* [46], de Hosseini *et al.* [82] [83], de Heidergott [79], Sandve et Aven [133]. Par ailleurs, outre les modèles de réparation minimale, d'autres approches sont proposées en tenant compte de la structure des systèmes à maintenir. Il y a notamment les problèmes de regroupement des tâches de maintenance, surtout pour des systèmes nécessitant d'importants coûts logistiques entre les différentes tâches.

2.5 Problème de regroupement des activités de maintenance

Pour des systèmes complexes de grande taille comprenant différents composants, le traitement de chaque composant individuellement nécessite des données propres, données dont la collecte est difficile comme dans la majorité des cas de données relatives à la maintenance. Mais au delà des difficultés relatives à chacun des composants, leur appartenance à l'ensemble du système entraîne d'autres genres de difficultés. En effet, dans ce genre de situation, la performance globale du système est beaucoup plus envisageable que les problèmes individuels de chaque composant.

Une activité de maintenance entreprise sur un système technologique de ce genre implique souvent d'importantes contraintes logistiques (et donc des coûts logistiques) relatifs au lien entre les différentes parties du système. Ces coûts logistiques (ou de façon équivalente ces temps logistiques) sont souvent appelés "set-ups" dans la littérature. Ils peuvent être par exemple le coût de préparation associé à l'ouverture d'une machine [161], [169], [170]. Ils peuvent aussi être le coût dû à la perte de production pendant le temps que dure la maintenance si le système ne peut être utilisé en ce moment, ou encore bien autre chose, à des déplacements des réparateurs entre les différents sites du système si celui-ci est distribué. Le coût (temps) logistique dépend donc de

l'architecture du système et peut-être lié à plusieurs activités entreprises sur ce système. Le regroupement (classification) des activités permet d'amoindrir certains de ces coûts en profitant du fait que l'exécution d'une classe d'activités peut exiger un seul coût logistique au lieu de coûts logistiques individuels à chacune des activités. La classification peut, par exemple, consister en l'exécution simultanée de plusieurs activités. Ces stratégies permettent d'optimiser les ressources mises en œuvre. Parmi les travaux qui ont abordé la classification des activités de maintenance sous un aspect ou sous un autre, il y a notamment les travaux de Van Dijkhuizen et Van Harten [161], de Van Harten [162], de Haurie et L'Ecuyer [77], de Seshadri [137], de Seshadri et Pinedo [138], de Wilderman *et al.* [170, 169], de Chakravarty *et al.* [35], de Dekker *et al.* [48, 51, 55], de Assaf et Shantikumar [14], et de Sridhar [145].

Dans [161], Van Dijkhuizen et Van Harten posent le problème de classes d'activités de maintenance en se basant sur la fréquence des activités et considèrent la fréquence d'une classe d'activités comme le maximum des fréquences des activités de cette classe. Cela leur permet de calculer le coût moyen par unité de temps des activités d'une classe comme produit de la fréquence et du coût total des activités de cette classe, et ensuite de minimiser ce coût à long terme. Pour cela, ils ont développé quelques règles de dominance qu'ils ont utilisées dans des algorithmes par programmation dynamique. Pour Dekker et Roelvink [51], ils partent du concept de coût marginal pour déterminer la date optimale pour le remplacement préventif d'un groupe de composants. Ils comparent leur approche à des politiques de remplacement par bloc qui remplacent le groupe de composant à des intervalles fixes. Les coûts à minimiser sont des coûts moyens à long terme. De plus, dans [51] Dekker propose une approche pour la combinaison des travaux de maintenance préventive suivant les priorités à ces travaux.

Seshadri et Pinedo [138] quant à eux considèrent le coût des actions correctrices (maintenance) comme le coût de la fourniture de service de production. Sur cette base, ils supposent que les activités de maintenance sont groupées en classes. Chaque classe de travaux se caractérise par son itinéraire dans le système, ses exigences opératoires et sa priorité. Les tâches de chaque classe arrivent suivant un processus stochastique sur des centres de machines et constituent des files d'attente au niveau de ces machines. Le problème considéré est d'allouer de façon optimale les ressources aux centres de machines afin de minimiser les "tâches en attente" au niveau des machines. Pour ce problème, ils ont proposé des algorithmes qui combinent optimisation et simulation. Ces travaux font suite aux travaux de Seshadri [137].

En utilisant aussi de la programmation dynamique, Wilderman *et al.* [170], proposent une politique dynamique pour le regroupement des activités de maintenance. Dans leur approche, ils considèrent que les fréquences à long terme d'activités de maintenance regroupées en classes, résultent des décisions dans le court terme, compte tenu des informations disponibles. Leurs travaux se résument en cinq étapes :

1. Une phase de décomposition qui formule un modèle à horizon infini pour chaque activité séparément et optimise le modèle afin d'avoir la fréquence optimale par rapport aux coûts moyens à long terme.

2. Une phase de pénalisation de chaque activité, consistant à leur attribuer des fonctions de pénalité.
3. Une tentative de planification, basée sur les règles individuelles de la phase 1.
4. Une phase de regroupement des activités, en introduisant des reports dans les tentatives de planification à la phase 3, afin de permettre des exécutions conjointes des tâches. Ils montrent que sous certaines conditions, un regroupement optimale peut être obtenu à ce niveau.
5. Enfin une phase de fixation de l'horizon de travail. Si à cette phase les décisions ne sont pas satisfaisantes, on retourne à la phase 3.

Dans les travaux de Seshadri et Pinedo, il est question d'allouer des ressources à des activités de maintenance sur des machines dans des centres de maintenance. Par ailleurs, Kool [95] aborde un problème où des groupes de machines interviennent aussi, mais tout en mettant l'accent sur le côté allocation de réparateurs (nous en reparlerons dans la section suivante). Les problèmes spécifiques d'allocation de réparateurs est l'un des problèmes importants. C'est l'objet de la section qui suit.

2.6 Problèmes d'allocation de réparateurs

Qui traite de maintenance est confronté au problème de ressources indispensables à l'accomplissement de cette fonction. Parmi ces ressources, figurent en bonne place les agents responsables des actions de réparation (les réparateurs). Le problème d'allocation de réparateurs est l'un des problèmes incontournables en maintenance. Les principaux travaux consacrés au problème d'allocation de réparateurs pour la maintenance sont notamment ceux de Derman *et al.* [58], de Frostig [68, 69, 70], de Kool [95], de Righter [130], de Seshadri et Pinedo [138], de Katehakis *et al.* [89, 90], de Smith [144], de Wartenhorst [166]. La plupart des méthodes développées considèrent des modèles stochastiques.

Kool [95] considère un problème dans lequel m groupes de composants identiques fonctionnant en parallèle. Les composants de chaque groupe ont un temps moyen avant défaillance distribué suivant la loi exponentielle. Sur les m groupes, k doivent être en marche (au moins un composant dans chacun des k groupes) pour que le système fonctionne. Un seul réparateur est disponible pour assurer les réparations de tout le système. Ils ont proposé une approche suivant laquelle le réparateur est alloué au groupe ayant le plus petit nombre de composants en marche. Il appelle cette approche la politique du plus petit groupe (SGP - Smallest Group Policy). Et comme objectif, la politique SGP cherche à maximiser le temps moyen de bon fonctionnement du système. L'affectation du réparateur se fait suivant un processus de Markov. Il a proposé des méthodes de programmation dynamique pour la résolution. Comme nous l'avons mentionné dans la section précédente, l'approche proposée par Seshadri et Pinedo [138] est aussi suivant un processus stochastique et se base sur les files d'attente. Frostig [70] a aussi considéré le problème d'allouer un seul réparateur à la réparation de m machines. Le travail cherchait conjointement à allouer le réparateur et à assurer un contrôle optimal du taux de service. Le temps inter-défaillance et la durée de réparation sont des variables aléatoires suivant la loi exponentielle et le coût était une fonction continue et convexe. Elle a alors montré quelques propriétés de monotonie pour la politique optimale.

Notons qu'avant Frostig, Smith [144] a étudié un problème de m composants en série qui sont maintenus par un seul réparateur. Le temps inter-défaillance et la durée de réparation de chaque composant sont des variables aléatoires suivant la loi exponentielle. Son objectif était la fiabilité du système. Il a montré que pour un nombre m égal à deux composants, la politique SFR (Smallest Failure Rate - politique dans laquelle le composant servi est celui qui a le plus petit taux de défaillance) permettait de maximiser la disponibilité du système. De leur côté, Derman *et al.* [58] ont considéré un cas où les taux de défaillance sont identiques pour tous les composants et ont étendu le résultat de Smith au cas général de m composants. Katehakis et Derman [89] ont généralisé le résultat de Smith au cas où les taux de défaillance des composants sont différents. Le problème à un seul réparateur a été aussi considéré par Righter [130] qui avait pour objectif de minimiser stochastiquement le nombre de composant en marche.

Pour les cas de plusieurs réparateurs, nous avons notamment les travaux de Frostig [68], de Wartenhorst [166] et de Katehakis et Melolidakis [90]. Ces derniers (Katehakis et Melolidakis) ont considéré une première généralisation de l'approche de Smith avec plusieurs réparateurs travaillant à des vitesses inégales. Ils ont montré que le fait d'allouer le composant le plus fiable au réparateur le plus rapide permet d'optimiser stochastiquement le nombre de composants en marche. Une extension de cette première généralisation a été proposée par Frostig [68]. Dans le travail de Frostig [68], m réparateurs doivent assurer la maintenance de n machines en série. Lorsqu'une machine tombe en panne, elle est affectée à un des réparateurs ou attend dans une file d'attente la disponibilité d'un réparateur. La préemption est permise dans les tâches de réparation et donc un réparateur peut être ré-affecté à tout moment. L'objectif est d'allouer les réparateurs aux machines en panne, de façon à maximiser le temps moyen de bon fonctionnement du système ou, de façon équivalente, à minimiser la durée moyenne d'indisponibilité (MDT - Mean Down Time). L'approche qu'elle propose se place dans le cadre markovien. Pour Wartenhorst [166], son approche se base sur la théorie des files d'attente. Il considère un système de files d'attente multiples constitué d'un nombre N de stations parallèles à un serveur, chaque station ayant une capacité de file d'attente infinie. Les serveurs sont sujets à des pannes et chaque serveur en panne est réparé par l'un parmi un ensemble de K réparateurs parallèles. Il s'intéresse particulièrement à une des stations (la station 1) et étudie l'influence des pannes de serveurs et des capacités de réparation finie (K) sur la taille de la file à cette station à long terme. Il considère que cette station fonctionne suivant une file d'attente $M/M/1$. Les arrivées sur cette station se font suivant un processus de Poisson et les services sont suivant le principe du premier venu premier servi (FCFS). Les durées inter-défaillance de cette station sont indépendantes et identiquement distribuées suivant une loi exponentielle. Il montre quelques propriétés de la taille exacte de la file à cette station.

Comme on peut le noter dans tous ces travaux, les approches proposées sont stochastiques et l'objectif à optimiser est souvent l'espérance mathématique d'un paramètre. Les approches varient autour des chaînes de Markov, des files d'attente pour l'allocation des réparateurs. Pour le cas d'un seul composant, il y a aussi les processus de renouvellement alterné qui prennent aussi en compte des pannes aléatoires et des durées de réparations aléatoires du composant mais, dans ce cas, le problème du réparateur n'est pas particulièrement abordé. Des approches, autres que celles

énumérées ci-dessus peuvent être particulièrement intéressantes pour la gestion de la maintenance. C'est le cas notamment de quelques travaux en gestion de production comme nous allons le voir dans la section qui suit.

2.7 Planification de la maintenance et gestion de production

D'un point de vue de production, les activités de maintenance préventive et curative peuvent être considérées comme des tâches qui doivent alors être planifiées et ordonnancées au même titre que les tâches régulières de production. De ce point de vue, la planification intégrée de la maintenance et de la production semble tout à fait naturelle. Au delà des travaux de Seshadri [137] et Seshadri et Pinedi [138] que nous avons cités plus haut, qui prennent la maintenance sous un aspect de production mais en adoptant des approches stochastiques, d'autres travaux sont purement déterministes.

Weinstein et Chung [167] ont proposé une approche pour l'intégration des politiques de maintenance dans les prises de décision en production. Leur objectif est de minimiser les coûts que cette politique engendre et les déviations de la production par rapport au niveau dans une politique de production sans maintenance. Leur approche est constituée de trois étapes et se base principalement sur un modèle de programmation linéaire en nombres-mixtes (utilisée dans les étapes une et deux), et la simulation par événements discrets (utilisée dans la troisième étape). Ils ont adopté une hypothèse de maintenance parfaite pour les tâches de maintenance préventive. De leur côté, Qi *et al.* [127] ont considéré un problème d'ordonnancement de tâche sur une machine sous l'hypothèse que la machine n'est pas disponible de façon permanente et que des opérations de maintenance préventive qui interviennent sont intercalées dans les opérations de production. Leur objectif est de minimiser la somme des dates de fin des tâches (équivalente au temps de réponse - temps de séjour total) de toutes les activités. Ils ont montré que le problème est NP-difficile et ont proposé une heuristique qui se base sur la règle SPT (shortest processing time) et une procédure de branch-and-bound pour sa résolution. De la même façon, Lee et Chen [102], ont étudié l'ordonnancement de tâches sur machines parallèles avec la contrainte que chacune des machines devra subir une action de maintenance préventive avant la fin des travaux. Ils avaient aussi pour objectif de minimiser la date de fin pondérée totale des tâches (équivalente au temps de réponse pondéré des tâches). Ils ont procédé suivant deux cas différents. Dans le premier cas, ils supposent qu'il y a assez de ressources pour que toutes les machines puissent subir simultanément les interventions de maintenance. Dans le deuxième cas, ils supposent qu'une seule machine à la fois peut subir une intervention de maintenance. Ils ont établi des propriétés donnant la date de fin au plus tard des tâches avant les tâches de maintenance et la date de fin au plus tard des tâches après les tâches de maintenance sur chaque machine. Ils ont montré que le problème est NP-difficile même lorsque les tâches sont identiquement pondérées et ont proposé une méthode de branch-and-bound basée particulièrement sur une formulation du problème en un problème de partition et en utilisant la génération de colonnes.

Graves et Lee [74] quant à eux ont étudié un problème sur une seule machine avec pour objectif la minimisation de la date de fin pondéré des tâches d'une part, et, d'autre part, la minimisation

du retard maximum des tâches sur une machine avec l'hypothèse que des opérations de maintenance sont intercalées dans la production. La machine est sujette à des actions de maintenance qui doivent avoir lieu sur un horizon de temps donné. Dans leurs problèmes, une tâche commencée qui n'a pu être complétée avant l'intervention de maintenance doit alors subir à la reprise un temps de préparation additionnel. Ils ont montré que le problème est NP-complet et dans une certaine configuration, ont proposé une méthode de programmation dynamique pour la résolution. Dans une autre configuration, ils ont proposé des heuristiques basées, d'une part, sur la règle SPT (shortest processing time) pour la minimisation de la date de fin totale et, d'autre part, sur la règle EDD (earliest due date) pour le retard maximum. Dans ce même registre, Bennour *et al.* [22] ont proposé une approche basée sur la théorie des graphes pour un problème d'intégration des politiques de maintenance dans le cycle de fonctionnement d'une machine. Dans la plupart de ces approches, les durées des tâches de maintenance sont représentées par leur MTTR (temps moyen de réparation). Murthy et Asgharizadeh [115] eux abordent le problème de tâche de maintenance plutôt sous un angle de théorie de jeux.

Outre les problèmes de gestion des tâches de maintenance, certains travaux ont aussi abordé le problème de la gestion de stocks en maintenance. Dans [12], Anily *et al.* abordent un problème de gestion des pièces de rechange d'un ensemble de m machines nécessitant chacune des pièces différentes. Ils étudient les stocks de rechange de chacune des machines. Ils montrent qu'il existe une politique cyclique de gestion des stocks de maintenance. Leur objectif est de minimiser le coût moyen de cette politique à long terme. Ils montrent une borne supérieure et deux bornes inférieures à la solution du coût et proposent deux heuristiques. D'autres problèmes de ce type sont étudiés par Hwang [84], Alfredsson [11], Sarker et Haque [134], Van der Duyn Schouten et Vanneste [160]. Une particularité du travail réalisé par Van der Duyn Schouten et Vanneste [160] est de considérer ce problème avec une approche markovienne. Bien qu'abordant un problème qui se rapproche intimement de la gestion de production, l'approche est basée sur les processus stochastiques. Ils analysent un modèle dans lequel la décision de commencer une intervention de maintenance préventive ne dépend pas seulement de la condition de la machine, mais aussi du contenu du buffer correspondant. Hauri [76] aborde un problème similaire avec toujours les processus stochastiques d'un point de vue de contrôle optimal. Il propose une approche basée sur la décomposition par rapport à l'échelle du temps en utilisant le fait que le changement de condition d'une machine a lieu sur une échelle de temps plus grande que les changements dans l'inventaire des produits finis de pièces de rechange. Hauri et Van Delft [78] continuent dans la même voie.

D'autres directions de recherche sont abordées dans la littérature, mais compte tenu de leur proportion faible dans la littérature, nous n'en ferons pas de section entière. Nous en parlerons néanmoins brièvement dans la conclusion de ce chapitre.

2.8 Conclusion

Les objectifs de la maintenance, aussi bien sur le plan de la conception, de la modélisation que de l'organisation sont de réduire et, si possible, optimiser différents coûts relatifs à l'état des systèmes,

d'abord en phase de conception, puis en phase d'exploitation, voire éventuellement de démantèlement. Les objectifs sont entre autres, de minimiser les coûts dus à des pannes qui sont imprévisibles et donc inévitables mais dont l'occurrence peut être réduite au mieux ; les coûts dus à l'exploitation en états dégradés (états critiques) des systèmes ; les coûts dus à des pertes de production en cas de maintenance non entreprise à des dates données ; les coûts dus à une maintenance préventive entreprise trop tôt ou trop tard (fréquences trop élevée ou déficit de maintenance) pour ne citer que ceux-là. Le dernier coût est dû à des avances ou des retards des interventions de maintenance préventive par rapport à des dates requises.

La modélisation et l'optimisation de la maintenance pour un management adéquat sont devenus un domaine essentiel de la recherche. Ceci est mis en évidence par la très grande attention portée ces dernières années à l'ensemble des mesures préventives pour une qualité totale efficace. L'attention portée à la modélisation stochastique et à l'analyse du management de la maintenance est intimement liée à la recherche intensive et à l'évolution rapide dans les domaines voisins tels que la théorie des files d'attente, la théorie Markovienne en particulier et tous les processus stochastiques en général, et les différentes théories en gestion de la production pour le soutien logistique. La pratique industrielle a clairement reconnu l'importance capitale de la maintenance comme mesure préventive et ce n'est pas le nombre et la quantité des expéditions de maintenance et de l'ensemble des mesures préventives qui contrediraient ce fait. Cependant, comme l'ont souligné Van der Duyn Schouten et Tapiero [159], le nombre de modèles stochastiques qui ont eu du succès en application est plutôt limité. Plusieurs causes justifient cela. Par exemple, l'obtention de données fiables nécessaires à la maintenance ne peut être réalisée en un court laps de temps. Les données doivent faire l'objet de compromis entre plusieurs parties directement ou indirectement liées au processus de maintenance. Par ailleurs, la valeur réelle de la maintenance se trouve dans des événements qui se produisent rarement (comme une machine qui ne tombe pas en panne), ce qui rend l'obtention de ces données, au mieux, des tâches spéculatives. Pour l'ensemble de ces raisons, le développement de modèles qui procurent une bonne compréhension pour un bon management de la maintenance, a toujours été d'une grande importance.

2.8.1 Principales observations de l'état de l'art

De l'observation de la littérature, plusieurs constats se font. Ces constats vont de la nature des politiques de maintenance, à leur mise en application en passant par la nature du système abordé, le point de vue adopté pour la proposition d'une approche de résolution d'un problème de maintenance, les différentes hypothèses sur lesquelles se basent les approches, etc.

Les systèmes étudiés sont en général des systèmes simples, constitués le plus souvent d'un nombre de composants dépassant rarement trois. Les objectifs visés consistent à optimiser des paramètres (souvent des espérances mathématiques de ces paramètres) sur un horizon le plus souvent infini. L'étude de caractéristiques asymptotiques est souvent de mise dans ces cas. Compte tenu de la grande variété des problèmes de maintenance abordés, les méthodes de résolution sont aussi

variées que les problèmes. Plusieurs modèles de maintenance considèrent la classification des activités de maintenance sur la base d'un horizon à court terme, à long terme, voire tendant vers l'infini. Les modèles à long terme se basent sur les propriétés à court terme. Certaines méthodes proposées utilisent des modèles statiques, se basant sur la fréquence des activités à entreprendre. D'autres, par contre, proposent des politiques dynamiques. Ces dernières permettent une planification itérative des activités de maintenance.

La plupart des modèles dans la littérature proposent la détermination de périodes optimales pour les tâches de maintenance préventive (systématique et conditionnelle) afin de satisfaire des exigences de fiabilité et de disponibilité des systèmes, et ce, à moindre coût. Lorsque le problème présente un caractère stochastique, l'espérance mathématique d'un paramètre vient en général remplacer le coût. En général, ces méthodes font des hypothèses très fortes notamment sur la disponibilité des ressources de maintenance ; ces ressources sont en général supposées instantanément disponibles au pied des machines et aussitôt qu'on en a besoin ; les équipements sont en général supposés réparés instantanément. Ces hypothèses sont souvent trop fortes pour les conditions pratiques dans la réalité. Dans la pratique, il est beaucoup plus difficile de faire face à un ensemble tout entier d'équipements réparables avec un nombre limité de techniciens. En effet, un réparateur n'est en général pas disponible au moment théoriquement prévu. En plus, des imprévus peuvent repousser les tâches de maintenance à des dates ultérieures, contribuant ainsi à davantage de dégradations des entités et accroissant le coût de la maintenance. Par ailleurs, les systèmes soumis aux actions de maintenance sont de plus en plus composés d'installations distribuées sur différents sites, avec l'exigence de les maîtriser au mieux.

Notons que le coût de gestion des opérations de maintenance dépend des caractéristiques de fiabilité et de maintenabilité des composants des systèmes tels que définies lors de la conception, du nombre de ressources (réparateurs, techniciens de maintenance) disponibles pour les actions de maintenance, de leurs qualifications et de la stratégie de maintenance pratiquée. Il dépend aussi des différents coûts logistiques qui sont liés à l'aspect distribué des systèmes en question, et de la plus ou moins grande réactivité avec laquelle les opérations de maintenance seront conduites, réactivité notamment due à des approches de prises de décision en temps-réel dans le processus de maintenance. Ainsi, il y a des coûts dus aux conditions de fonctionnement ou de panne des systèmes et des coûts dus à la logistique mise en œuvre dans la politique de maintenance.

Ce travail a ainsi pour but de répondre à un certain nombre de ces préoccupations. Afin de bien se situer par rapport à ces préoccupations prises en compte dans ce document, nous allons dans la sous-section suivante positionner ce travail par rapport aux différents problèmes ci-dessus mentionnés et définir les objectifs visés.

2.8.2 Objectifs de nos travaux

Le but de ce travail est de contribuer au développement d'approches pour répondre à quelques-unes des préoccupations ci-dessus mentionnées avec pour objectifs d'aider à la prise de décision en temps-réel dans le processus de maintenance. Les objets sur lesquels reposent nos préoccupations

sont les systèmes distribués multi-sites, avec des contraintes logistiques non négligeables entre les différents sites et partageant les mêmes ressources de maintenance entre ces sites. Ces ressources sont limitées devant la taille du système et doivent être utilisées suivant leur disponibilité, leur capacité et au mieux suivant les besoins du système. Chaque site du système comprend des entités réparables. Contrairement à des hypothèses habituelles de la littérature, les durées des tâches de maintenance ne sont pas négligeables. Elle dépendent des caractéristiques des différentes entités du système et sont relatives à leur MTTR.

Les problèmes auxquels nous nous intéressons vont dans le sens des exigences de planification et d'ordonnement des tâches de maintenance et d'affectation en temps-réel des ressources suivant leur disponibilité pour la réalisation de ces tâches afin de maîtriser certains coûts inhérents au fonctionnement du système, en réduisant les délais des interventions de maintenance et en équilibrant au mieux les charges de maintenance sur les ressources utilisées. Parmi les différents types de coûts que nous abordons dans ce travail se situent notamment ceux qui suivent.

- Les coûts des états critiques des entités du système. Ces coûts d'états critiques sont dus à des fonctionnement en états de dégradation avancées des entités ou à des pannes en cas de réactivité faible face à ce problème. Il y a une correspondance entre les coûts des états critiques et les temps de réponse (ou encore temps de séjour) des tâches de maintenance.
- Les coûts dus à des déficits (fréquence faible) de maintenance. Ces coûts sont notamment relatifs à des retards de tâches de maintenance. Ils peuvent être envisagés seuls, ou conjointement avec les coûts des états critiques.
- Les coûts dus à des fréquences élevées d'interventions de maintenance, entraînant donc de la sur-maintenance. Ils sont dus particulièrement à des avances de tâches de maintenance par rapport à des dates requises. Un juste équilibre est de considérer conjointement ces coûts de fréquences élevées de maintenance avec les coûts de déficit de maintenance. Cela engendre donc des coûts dus à des avances et des retards des tâches de maintenance.

Les problèmes peuvent aussi se poser en termes de combinaison linéaire ou convexes de ces différents coûts dans le but d'atteindre des objectifs spécifiques et cela dépend des facteurs de pondération de ces différents coûts. Ces coûts sont en général fonction de la disponibilité, cette dernière étant révélatrice des dégradations des entités constituant le système.

Pour satisfaire ces exigences, différents problèmes doivent être résolus dans le but de maintenir la disponibilité du système au dessus d'une limite minimale, de minimiser le coût des opérations de maintenance et d'affecter en temps-réel les tâches aux ressources de maintenance, le tout dans un processus optimal de décision. Ce processus doit tenir compte de différentes contraintes : dates d'arrivée inégales des tâches, temps logistiques non négligeables entre les sites, séquences optimales de tâches d'une intervention, précédences entre les tâches d'une même entité et critères de priorité et d'urgence de chaque tâche. La plupart de ces contraintes, surtout celles relatives aux dates d'arrivées des tâches de maintenance dans le système et aux temps logistiques non négligeables et dépendant de la séquence des tâches de maintenance, rendent particulièrement délicat

le processus de décision ici abordé. Mais nous tirerons avantage de quelques développements analytiques dans ce document. Dans cette démarche, les caractéristiques et lois de comportement en fiabilité et en maintenabilité et les sollicitations d'intervention sur les entités sont connues.

Pour résoudre les problèmes ainsi posés, nous proposons des approches mettant en œuvre des méthodes d'ordonnancement d'activités (tâches ou actions) de maintenance basées sur des règles de priorité dont nous montrons l'optimalité locale dans ce document. Partant d'une ressource de maintenance, ces règles de décision s'inspirent des principes d'ordonnancement sur une seule machine. Elles sont utilisées dans des algorithmes d'aide à la décision en temps-réel pour la planification dynamique des tâches de maintenance des systèmes distribués. Aux solutions fournies par la plupart de ces règles nous avons déterminé des bornes inférieures.

Chapitre 3

Maintenance avec coûts impliquant des temps de réponse

3.1 Introduction

Ce chapitre est consacré à la première des préoccupations mentionnées dans les objectifs. Cette préoccupation est relative aux coûts des états critiques dus au fonctionnement en état de dégradation avancée du système ou à des pannes fortuites qui surviennent au mauvais moment au mauvais endroit. Ces coûts d'états critiques, comme nous allons le voir dans la spécification de la position du problème, se rapportent à des temps de réponse des tâches de maintenance. Ici, nous utilisons le terme "temps de réponse" plutôt que "temps de séjour" pour désigner le terme générique "flow-time" des tâches de maintenance. Cette appellation est mieux adaptée à la situation présente et elle permettra par ailleurs d'avoir à l'esprit que ces coûts sont très dépendants de la plus ou moins grande réactivité avec laquelle le système de décision répond aux sollicitations des entités à maintenir.

Les spécifications et les caractéristiques de ce problème seront détaillées dans la section 2 qui précise la position du problème objet de ce chapitre. Cette section nous permettra de bien définir le problème dont il est question. Dans cette position du problème, nous ressortirons les différentes propriétés du problème. Nous verrons alors que suivant la configuration du système, le problème des coûts d'états critiques peut se poser sous différents aspects avec des niveaux de difficulté différents dans la prise de décision. C'est ainsi que dans certaines configurations, les hypothèses sur les contraintes logistiques entre les sites peuvent être simplifiées, et pas dans d'autres. Par ailleurs, il y a des coûts spécifiques à chaque entité et ces coûts peuvent engendrer des facteurs de pondération de ces entités. Dans certains cas, les poids des entités peuvent être égaux et pas dans d'autres. Compte tenu de tout cela, quatre sous-problèmes du problème général des coûts des états critiques, avec des contraintes logistiques entre les sites, seront définis. Chacun de ces sous-problèmes sera considéré dans une section séparée. Ainsi dans la section 3, nous allons considérer ce problème dans sa version la plus simple avec des temps logistiques faibles entre les sites et des entités également pondérées dans le système. Dans la section 4, nous traitons le problème avec des

entités inégalement pondérées et des temps logistiques faibles entre les sites. Ensuite, les temps logistiques non négligeables seront introduits dans la section 5 mais avec des entités également pondérées. Nous considérerons finalement la configuration la plus générale avec des entités inégalement pondérées dans le système et des temps logistiques non négligeables entre les sites dans la section 6 avant de tirer les principales conclusions de ce chapitre dans la section 7. Dans chacune de ces configurations, les approches proposées seront illustrées sur des exemples simples avant leur utilisation sur des systèmes de taille réelle et dans des conditions de fonctionnement réelles.

3.2 Position et modélisation du problème

Le problème qui est abordé dans ce document est celui de la modélisation d'un processus de maintenance et de la recherche de techniques appropriées pour l'allocation des ressources nécessaires à la réalisation des opérations de maintenance d'un système industriel distribué sur différents sites. L'étude porte sur un système ou groupe formé d'un nombre donné de sites, chaque site étant composé d'entités réparables. Le système dispose de ressources de maintenance dont certaines sont utilisées en commun par l'ensemble des sites. Il faut faire une allocation optimale des ressources suivant leur disponibilité (pas au sens sûreté de fonctionnement) pour des opérations de maintenance - préventive dans un premier temps, puis curative - à tout moment, sur des horizons donnés. Sur chaque horizon, les objectifs primordiaux seront de garder le système en état nominal de fonctionnement efficace, tout en minimisant les coûts qu'engendre la réalisation de cet objectif sur ces horizons.

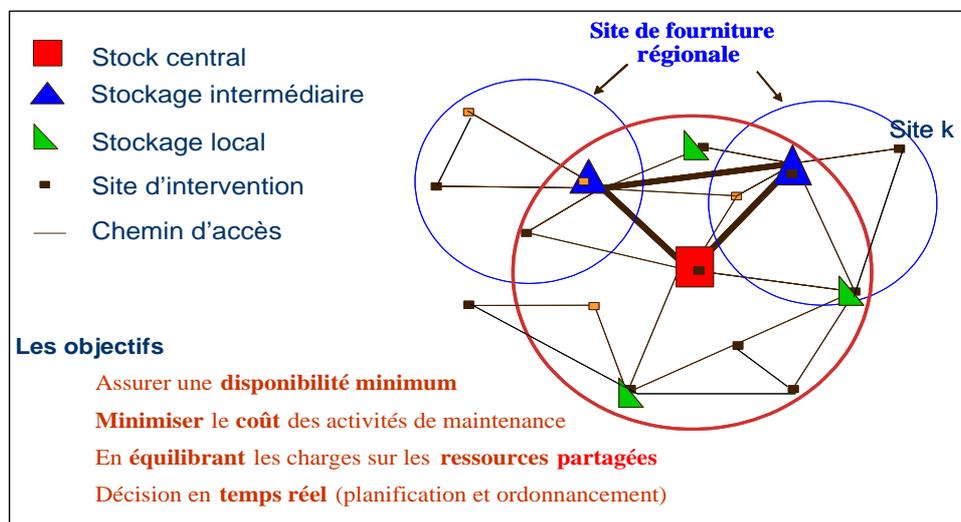


Figure 3.1 – Structure organisationnelle d'un groupe à soutenir par des opérations de maintenance

Les sous-systèmes du système étudié possèdent des entités (équipements) qui nécessitent en réalité des ressources de maintenance de différentes classes (ressources humaines, pièces de rechange, équipements d'intervention, services temporels, etc.). Mais dans le modèle que nous proposons ici, nous considérons sans perte de généralité une seule classe de ressources pour l'ensemble du système. Nous reviendrons dans la conclusion sur la justification de ce choix. Le type

de ressources considéré est humain. Dans les modèles que nous proposons, nous considérons que les ressources humaines de maintenance (réparateurs) ont des performances identiques et toutes les ressources peuvent être utilisées de façon équivalente sur chaque entité. Dans la suite donc, le terme entité désignera les équipements réparables sur les sites du système. Le terme ressource désignera l'une des ressources humaines utilisables de façon équivalente et en parallèle sur les différentes entités.

Nous supposons que les sites du système fonctionnent de façon indépendante, c'est-à-dire qu'une panne sur un site quelconque n'influence pas le fonctionnement des autres. Par contre, nous supposons que les entités d'un site donné fonctionnent en série. Ainsi, pour qu'un site soit disponible, il faut que toutes les entités de ce site soient en état de bon fonctionnement. Ces hypothèses se basent sur les structures que l'on trouve en général dans les systèmes industriels distribués où, sur un site donné, les entités sont interdépendantes, mais où l'on observe en revanche une certaine autonomie d'un site par rapport à un autre tout en ayant souvent un objectif de performance globale du système formé par l'ensemble des sites.

Avant de passer aux approches cherchant à optimiser les objectifs du système de gestion de maintenance que sont la performance du système (au niveau de sa disponibilité) et le coût de réalisation de ce premier objectif, nous allons d'abord voir quelles mesures de modélisation nous adoptons. Nous utiliserons dans la modélisation et la résolution de ce problème les notations suivantes :

Notations

- N est le nombre total de sites du système
- S_k est k^{ieme} le site du système
- N_k est le nombre d'entités du site S_k
- \mathcal{E}_{ik} est l'entité i du site S_k
- λ_{ik} est le taux de défaillance de l'entité \mathcal{E}_{ik} (pas nécessairement constant)
- μ_{ik} est le taux de réparation de l'entité \mathcal{E}_{ik}
- p_{ik} est la durée d'une tâche de maintenance sur l'entité \mathcal{E}_{ik}
- $A_{ik}(t)$ est la disponibilité de l'entité \mathcal{E}_{ik} à l'instant t
- M est le nombre total de ressources de maintenance dont dispose le système
- \mathcal{H} est l'horizon de planification
- $C_{tot}(\mathcal{H})$ est le coût total engendré sur l'horizon \mathcal{H} .

Nous introduirons d'autres notations dans la suite de ce chapitre telles que : α_{ik}^1 , β_{ik} , γ_{ik} , σ_{ik} , τ_{ik} , $r_{m,ik}$, $c_{m,ik}$, w_{ik} et s_{kl} . Ces notations seront expliquées au fur et à mesure qu'elles sont introduites.

Parmi les hypothèses dans la modélisation, le nombre total de ressources M est évidemment très inférieur au nombre total d'entités $\sum_{k=1}^N N_k$ dans le système ($M \ll \sum_{k=1}^N N_k$). Les tâches de maintenance (préventive et curative) sont supposées être exécutées sans préemption.

Les entités de chaque site fonctionnant en série, dès que l'un des composants est hors d'état de

fonctionnement, l'ensemble du site est indisponible. Ainsi, la disponibilité individuelle de chaque entité d'un site donné est indispensable pour la disponibilité de ce site. Pour cela, chaque entité de chaque site doit être le "moins indisponible possible". Pour ce faire, des contraintes sont imposées sur la disponibilité de chaque entité sur un site donné. Il faut se référer aux appendices A et B pour voir l'évolution de cette fonction disponibilité suivant le comportement de l'entité. La fonction disponibilité est en général une fonction décroissante du temps (en dehors de la phase de jeunesse de l'entité où cette fonction peut être croissante à cause notamment des réglages afin de fixer les équipements). La disponibilité A_{ik} de l'entité \mathcal{E}_{ik} devra rester au dessus d'un seuil α_{ik}^1 fixé (α_{ik}^1 donné entre $]0; 1[$). En fait, α_{ik}^1 est le seuil de disponibilité minimum requis pour l'entité \mathcal{E}_{ik} afin que le site \mathcal{S}_k fonctionne convenablement. Lorsque la disponibilité A_{ik} tombe en dessous du seuil α_{ik}^1 , on considère que l'entité \mathcal{E}_{ik} est en état critique de fonctionnement avec une forte probabilité de défaillance. L'objectif dans ce chapitre est de minimiser le coût des états critiques et de la réalisation de cet objectif sur l'horizon fini de planification \mathcal{H} . Nous nous intéressons à un horizon fini parce que dans la pratique, les systèmes sont en général considérés sur un horizon fini (mensuel, trimestriel, semestriel, annuel, quinquennal, etc.). À la fin de cet horizon, l'état du système est analysé, un inventaire est réalisé et l'état des lieux à la fin de chaque horizon sert de point de départ pour l'horizon suivant.

Il faut noter que cette façon de se fixer comme objectif un seuil à ne pas dépasser est un objectif bien réaliste qui, d'ailleurs, n'est pas nouveau. En effet, en se basant sur le concept de probabilité, pour les niveaux de sécurité et de fiabilité des avions, Pugsley ([163] [164]) dans ses travaux publiés en 1939 et 1942 par l'*Aeronautical Research Council de Londres* suggéra que le taux d'accident d'un avion, en considérant toutes les causes de panne susceptibles d'entraîner un accident, ne dépasse pas 10^{-5} par heure et 10^{-7} par heure pour les causes liées à la structure de l'avion. Cette suggestion de Pugsley sera prise en compte vers la fin des années 1960 par des applications dans la conception aéronautique.

Une entité \mathcal{E}_{ik} donnée qui fonctionne pendant un certain temps sans défaillance, verra sa disponibilité décroître au cours du temps vers le seuil α_{ik}^1 que nous appellerons désormais seuil critique. À la fin d'une intervention (tâche) de maintenance, nous supposons que cette entité est renouvelée, et que son état est considéré comme neuf (maintenance parfaite). Ainsi, une entité \mathcal{E}_{ik} mise en service (à neuf à l'instant initial) ou remise en service (à la suite d'une tâche de maintenance) à l'instant T , évolue vers un état critique atteint après une durée qui sera notée τ_{ik} pour cette entité. Les événements de ce type se rapportent à chaque entité et arrivent au cours du fonctionnement du système (sur l'horizon de planification) soit en phase de production, soit en phase d'intervention de maintenance, ce qui en fait un problème d'organisation en temps-réel (*on-line*). L'hypothèse de remise à neuf est une hypothèse généralement utilisée dans les modèles de maintenance. C'est le cas, par exemple, dans les travaux de Dekker *et al.* [53], [50], [54], [48], [51].

Si une entité \mathcal{E}_{ik} a un comportement avec une disponibilité exponentielle (taux de défaillance λ_{ik} et de réparation μ_{ik} constants) [163], [164], sa disponibilité à l'instant t sachant qu'elle est mise

ou remise en service l'instant T ($t \geq T$), est

$$A_{ik}(t - T) = \frac{\mu_{ik}}{\lambda_{ik} + \mu_{ik}} + \frac{\lambda_{ik}}{\lambda_{ik} + \mu_{ik}} \exp[-(\lambda_{ik} + \mu_{ik})(t - T)] \quad (3.2.1)$$

Et en considérant le seuil α_{ik}^1 , la durée τ_{ik} pourra s'exprimer à partir de l'expression 3.2.1 avec $T = 0$ comme suit.

$$\tau_{ik} = \frac{-1}{\lambda_{ik} + \mu_{ik}} \ln \left[\alpha_{ik}^1 \left(1 + \frac{\mu_{ik}}{\lambda_{ik}} \right) - \frac{\mu_{ik}}{\lambda_{ik}} \right] \quad (3.2.2)$$

Dans des cas plus généraux tels que la loi de Weibull (dont la loi exponentielle est un cas particulier), la disponibilité à l'instant t , après une mise ou une remise en service à l'instant T est $A_{ik}(t - T)$ où $A_{ik}(t)$ s'exprime comme ci-dessous. Les paramètres de la loi de Weibull pour l'entité \mathcal{E}_{ik} sont :

- γ_{ik} le paramètre de décalage ou point de départ des observations (nous le noterons γ en général)
- $\sigma_{ik} > 0$ le paramètre d'échelle de temps (nous le noterons σ en général)
- $\beta_{ik} > 0$ le paramètre de forme ou allure d'évolution (nous le noterons β en général).

Dans ces conditions, nous avons montré dans l'appendice B que si le taux de réparation $\mu : t \mapsto \mu(t)$ est intégrable sur tout sous-ensemble borné de \mathbb{R} , alors la fonction disponibilité $A(t)$ s'exprime pour tout instant $t > \max(0, \gamma)$ comme

$$A(t) = \exp \left(- \int_{\gamma}^t \mu(x) dx - \left(\frac{t - \gamma}{\sigma} \right)^{\beta} \right) \cdot \left\{ 1 + \int_{\gamma}^t \mu(x) \cdot \exp \left(\int_{\gamma}^x \mu(u) du + \left(\frac{x - \gamma}{\sigma} \right)^{\beta} \right) dx \right\} \quad (3.2.3)$$

Si le taux de réparation μ est constant, la fonction disponibilité $A(t)$ devient alors

$$A(t) = \exp \left(- \left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^{\beta} \right] \right) \cdot \left\{ 1 + \mu \sigma \cdot \int_0^{\frac{t - \gamma}{\sigma}} \exp(\mu \sigma x + x^{\beta}) dx \right\} \quad (3.2.4)$$

Dans ce dernier cas (lorsque μ_{ik} est constant) et lorsque le paramètre de forme β_{ik} est supérieur ou égal à 1, autrement dit dans les phases de fonctionnement de l'entité autre que sa phase de jeunesse, avec des réglages de fixation pour l'amener dans un état de fonctionnement normal ($\beta_{ik} \geq 1$ - le cas $\beta_{ik} = 1$ devenant le cas de la loi exponentielle), nous avons montré dans l'appendice B que la fonction disponibilité $A_{ik}(t)$ est strictement décroissante et tend vers une limite $A_{ik}(\infty)$ telle que $\left(0 \leq A_{ik}(\infty) \leq \frac{\mu_{ik}}{\mu_{ik} + \beta_{ik}/\sigma_{ik}} \right)$. Cette limite est décroissante en fonction du paramètre de forme β_{ik} et est égale à $\frac{\mu_{ik}}{\mu_{ik} + 1/\sigma_{ik}}$ lorsque $\beta_{ik} = 1$, ce dernier correspondant au cas de la loi exponentielle. Ces conditions sont celles du cas de fonctionnement stable (phase de maturité) et de la phase de vieillesse de l'entité. Ainsi pour une telle entité \mathcal{E}_{ik} , il existe une unique date τ_{ik} tel que $A_{ik}(\tau_{ik}) = \alpha_{ik}^1$. τ_{ik} peut être simulé à partir de la fonction disponibilité. Ici, l'atteinte d'un état de criticité est exprimée en fonction de la durée de fonctionnement. Elle peut être en rapport à d'autres paramètres. Par exemple, dans l'aéronautique, cela dépend en général du nombre d'heures de vol effectuées par un avion. Elle est aussi dans certains cas liée à une distance totale parcourue, à un nombre total de tours effectués, etc.

Au delà de la durée τ_{ik} , l'entité \mathcal{E}_{ik} est en état critique. La date d'occurrence d'un tel événement

pour la m^{ieme} fois sur l'entité \mathcal{E}_{ik} (après la m^{ieme} remise en service) sera noté $r_{m,ik}$. Si $c_{m,ik}$ désigne la date de la m^{ieme} remise en service (confondue avec la date de complétion de la m^{ieme} tâche de maintenance sur l'entité \mathcal{E}_{ik}), la relation entre les dates $r_{m,ik}$ et $c_{m,ik}$ est donnée par

$$r_{m+1,ik} = c_{m,ik} + \tau_{ik} \quad (3.2.5)$$

Le temps entre les dates $r_{m,ik}$ et $c_{m,ik}$ est un temps critique pour l'entité \mathcal{E}_{ik} . Chaque unité de temps que l'entité \mathcal{E}_{ik} passe dans un état critique a un coût unitaire égale à w_{ik} qui dépend de l'entité. Les coûts par unité de temps w_{ik} peuvent provenir de plusieurs facteurs dépendant de la situation des entités dans le système. Ils peuvent être dus à leur importance pour une fonction dans le système, leur sensibilité, etc. Dans la suite, nous appellerons ces coûts unitaires les poids des entités. Il faut ainsi minimiser le coût des états critiques du système sur l'horizon de planification \mathcal{H} . L'expression du coût total du système en état critique sur l'horizon \mathcal{H} est :

$$C_{tot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} w_{ik}(c_{m,ik} - r_{m,ik}) \right) \quad (3.2.6)$$

Chaque intervention (tâche de maintenance) sur l'entité \mathcal{E}_{ik} a une durée d'exécution p_{ik} qui est supposée, dans un premier temps, être constante et égale à son $MTTR$ $\left(\frac{1}{\mu_{ik}}\right)$. Le problème considérant le coût des états critiques et la disponibilité des entités du système sur l'horizon \mathcal{H} peut alors être formulé comme suit :

(\mathcal{P}_1) : Minimiser le coût $C_{tot}(\mathcal{H})$ (le coût des états critiques sur l'horizon \mathcal{H}) avec M ressources parallèles à performances identiques partagées par toutes les composantes du système, sachant que l'occurrence des événements critiques intervient à tout moment sur l'horizon, avec des entités (éventuellement) pondérées, et des temps logistiques (set-ups) entre les différents sites, tout en satisfaisant au mieux les conditions $A_{ik} \geq \alpha_{ik}^1$.

Le problème ainsi formulé est un problème d'organisation et de décision en temps-réel, puisqu'a priori, on ne peut établir exactement les dates d'occurrence des événements critiques sur l'horizon \mathcal{H} . Tout au plus, peut-on estimer les dates d'occurrence des premiers événements critiques. Ces dates d'occurrence d'états critiques dépendent des dates de remise en service des entités, ces dernières elles-mêmes dépendant de la disponibilité des ressources pour entreprendre les tâches de maintenance. Les temps critiques s'apparentent à des temps de réponse (flow-time) en ordonnancement de la production. Le *flow-time* est appelé *temps de séjour* ou *temps de réponse* en ordonnancement de la production. Nous allons adopter dans ce document le terme *temps de réponse* qui correspond bien à la situation présente. Le temps logistique (set-up) entre deux sites \mathcal{S}_k et \mathcal{S}_l sera noté s_{kl} . Ces temps logistiques (ou encore ces temps de préparation) dépendent de la séquence d'exécution des tâches de maintenance c'est-à-dire de la séquence dans laquelle chaque ressource passe d'un site à un autre. Les ressources (qui sont appelées machines en production) ont les performances identiques et le problème est un problème à ressources parallèles identiques. Sa résolution se fera en plusieurs étapes, suivant plusieurs configurations du système à savoir :

- Les entités ont des poids identiques et les temps logistiques sont faibles ; nous l'appellerons problème (\mathcal{P}_{11}) .

- Les entités ont des poids différents et les temps logistiques sont faibles ; nous l'appellerons problème (\mathcal{P}_{12}).
- Les entités ont des poids identiques et les temps logistiques sont importants ; nous l'appellerons problème (\mathcal{P}_{13}).
- Les entités ont des poids différents et les temps logistiques sont considérables ; nous l'appellerons problème (\mathcal{P}_{14}).

Mais il faut noter que dans tous ces cas de figure, nous avons la propriété suivante.

Proposition 3.2.0.1. *Le problème (\mathcal{P}_1) est NP-difficile dans les quatre cas de figure ci-dessus.*

En effet, prenons le problème dans sa forme la plus simple (\mathcal{P}_{11}) dans une situation statique (*off-line*) (où toutes les informations sur les tâches sont connues dès le début de l'organisation) et avec une ressource unique. Cela revient en ordonnancement de la production à un problème d'ordonnancement de tâches sur machine unique (single machine scheduling) de minimisation du temps de réponse total, avec des dates d'occurrence ou dates d'arrivée inégales des tâches ($n/1/r_i/\sum(C_i - r_i)$). Ce problème est NP-difficile (voir par exemple dans [41], [42] et [44]). Ainsi, le problème (\mathcal{P}_{11}) est NP-difficile. Les autres problèmes (\mathcal{P}_{12}), (\mathcal{P}_{13}), et (\mathcal{P}_{14}) dont le problème (\mathcal{P}_{11}) est un cas particulier sont donc aussi NP-difficiles.

Il faut noter qu'à première vue on serait tenté de penser que ce processus est un processus cyclique dans lequel les mêmes comportements du système se reproduisent sur des périodes identiques plus ou moins longues données. Mais en fait, cela n'est pas le cas. En effet, si l'on considère un système d'une entité ayant une ressource pour s'occuper de son processus de maintenance, cela pourrait se faire exactement aux moments où il le faut et on pourrait avoir un processus cyclique. Mais pour un système de plusieurs entités ayant un nombre réduit de ressources pour assurer le processus de maintenance, ceci devient quasi-impossible puisque tout dépend de la disponibilité (pas la disponibilité au sens sûreté de fonctionnement) des ressources ; une entité donnée peut être considéré à temps à un instant donné, et en retard à une autre date. Pour cela, dans toute la suite de ce chapitre et de ce document, nous regarderons le processus d'organisation de la maintenance en question comme non cyclique.

Les problèmes de décision en temps-réel sont souvent abordés sous deux formes. La première forme consiste à considérer le processus sous un aspect entièrement stochastique puis à étudier un comportement asymptotique ou étudier l'espérance mathématique d'un paramètre. La deuxième façon consiste, puisque l'on n'a pas d'idée précise de comment sera fait le futur, à considérer à chaque instant de décision, le processus dans la forme où il est à cette date précise. Cette deuxième option est celle qui est la plus utilisée. Elle a l'avantage de fournir à tout instant une option pour la prise de décision. L'une des premières approches utilisées pour les prises de décision en temps-réel est le *FIFO (First In First Out)* qui consiste à considérer les événements suivant leur ordre d'arrivée dans le système. Dans les applications en temps-réel, il est obligatoire de fournir une solution rapidement, sans pouvoir attendre les informations futures. La deuxième option permet de prendre en compte ce fait. Les premiers problèmes importants traités en temps-réel ont concerné le pilotage de systèmes complexes (avions, centrales nucléaire, etc.). Les problèmes en temps-réel sont

de plus en plus étudiés. On en rencontre notamment en ordonnancement des activités de radars [36], [61], en distribution de courriers, etc. Les applications en temps-réel, notamment en ordonnancement en temps-réel, se développent parce qu'elles permettent de réagir dynamiquement aux aléas de l'environnement. Les applications en temps-réel impliquent de développer des techniques rapides dans tous les cas de figure.

La suite de ce travail se fera suivant cette deuxième option. Nous considérerons donc à chaque date de décision un système localement statique (*off-line*) sur lequel sera basée une procédure de décision localement optimale pour le système global temps-réel. Ainsi, dans toutes les configurations considérées, nous procéderons dans un premier temps à une analyse *off-line* (sur le système localement statique) avant de continuer par une proposition de méthode en temps-réel.

3.3 Configuration avec des entités non pondérées et des temps logistiques faibles

Nous considérons dans cette section le problème (\mathcal{P}_1) sous sa forme (\mathcal{P}_{11}) c'est-à-dire dans la configuration où les différentes entités du système sont non pondérées (ou encore de même poids égal à 1) et les temps logistiques entre les sites peuvent être assimilés à 0. Nous sommes donc en présence d'un problème où les tâches, non pondérées, ont des dates d'occurrence inégales et où les temps logistiques sont faibles et assimilés à 0. Il faut donc minimiser le coût des états critiques qui est dans ce cas

$$C_{tot}^1(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} (c_{m,ik} - r_{m,ik}) \right) \quad (3.3.1)$$

Nous rappelons que ce problème est NP-difficile. De plus, dans le problème d'organisation en temps-réel les décisions se basent sur des choix locaux sur des problèmes localement statiques et nous allons considérer dans un premier temps un ensemble de n tâches (disponibles ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Chaque tâche i a une date d'occurrence r_i (qui ne sont pas égales pour toutes les tâches), une durée d'exécution p_i , et l'objectif est de minimiser leur temps total de réponse (ou temps total de séjour) non pondéré ($\sum_{i=1}^n (c_i - r_i)$ si la date de complétion de la tâche i est c_i). Nous considérons le problème sur une ressource et cherchons les options pour une décision garantissant une optimalité locale.

3.3.1 Le problème statique et les critères de décision

Les problèmes d'ordonnancement pour la minimisation du temps de réponse non pondéré des tâches ont été abondamment considérés dans la littérature aussi bien sur une ressource unique, sur des ressources parallèles qu'en flowshop. Mais cela n'étant pas l'objet principal de travail, nous laisserons le lecteur voir des analyses de la littérature dans [41], [44] et [42] par exemple. Pour l'aspect du problème qui nous intéresse dans ce travail et pour cette configuration-ci, nous allons considérer une propriété intéressante d'optimalité locale existant dans la littérature. Pour le

problème de minimisation du temps total de réponse non pondéré des tâches ordonnancées sur une ressource, avec des dates d'occurrence inégales des tâches telles qu'il est rappelé ci-dessus, Chu a défini la fonction ci-après et a montré le théorème qui suit. Dans ce problème, les tâches sont exécutées sans préemption et donc, une fois commencée, toute tâche est exécutée sans interruption jusqu'à sa complétion.

Définition 3.3.1.1. *Soit un ensemble de tâches devant être ordonnancées à un instant t donné sur une ressource. Chaque tâche i a une date d'occurrence (release date) r_i et une durée d'exécution p_i . Alors, on définit la fonction $PRTF$ (Priority Rule for Total Flow time) dont la valeur à l'instant t pour la tâche i est (voir dans [41], [44] et [42]) :*

$$PRTF(i,t) = 2 \max(r_i,t) + p_i \quad (3.3.2)$$

Sur la base de cette définition, il a été montré le théorème ci-dessous. Le lecteur peut trouver la preuve de ce théorème dans [44].

Théorème 3.3.1.2. *Soit un problème dans lequel deux tâches i et j sont à ordonnancer sur une ressource disponible après un instant t avec comme objectif de minimiser le temps total de réponse. On suppose que $PRTF(i,t) \leq PRTF(j,t)$. Soit alors π un ordonnancement dans lequel la tâche i est ordonnancée avant la tâche j et π' un autre ordonnancement ne différant de π que par l'inversion des tâches i et j . Alors, π domine π' . Autrement dit la solution fournie par π est meilleure que celle fournie par π' .*

Nous dirons, avec la définition qui sera introduite un peu plus loin, que la tâche i domine la tâche j à la date de décision t . La résolution du problème en temps-réel (\mathcal{P}_1) dans sa configuration (\mathcal{P}_{11}) sera donc basée sur cette fonction $PRTF$ et sur le théorème 3.3.1.2. Par ailleurs, la proposition suivante nous permettra par la suite de parler indifféremment des entités ou des demandes de tâches sur les entités.

Proposition 3.3.1.3. *En respectant le théorème 3.3.1.2 avec la fonction $PRTF$, on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} . En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, le théorème 3.3.1.2 permet d'ordonnancer la $m_1^{ième}$ tâche sur \mathcal{E}_{ik} avant la $m_2^{ième}$.*

Preuve. En effet, deux interventions quelconques sur l'entité \mathcal{E}_{ik} sont supposées avoir la même durée d'exécution p_{ik} qui est égale à son $MTTR \left(\frac{1}{\mu_{ik}} \right)$. De plus, les dates d'occurrence $r_{m_1,ik}$ et $r_{m_2,ik}$ sont telles que $r_{m_1,ik} < r_{m_2,ik}$. Donc $2 \max(r_{m_1,ik},t) + p_{ik} \leq 2 \max(r_{m_2,ik},t) + p_{ik}$. \square

Ainsi, dans la suite de cette section, nous parlerons indifféremment de sélection d'une tâche sur une entité ou de sélection de l'entité elle-même. Dans la procédure en temps-réel, nous utiliserons en plus de la fonction $PRTF$, du théorème 3.3.1.2 et de la proposition 3.3.1.3, la définition suivante.

Définition 3.3.1.4. *Soit t une date de décision. Nous définirons par $\Delta_{ik}(t)$ la durée depuis la dernière remise en service de l'entité \mathcal{E}_{ik} avant la date t .*

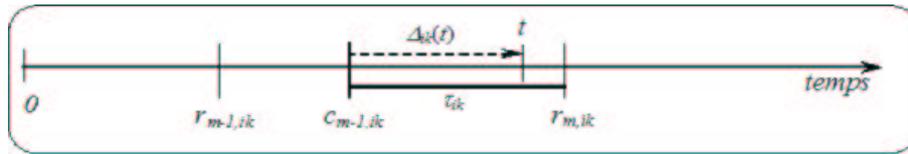


Figure 3.2 – Définition de $\Delta_{ik}(t)$

Nous dirons alors que la tâche de maintenance sur l'entité \mathcal{E}_{ik} est urgente à la date t si on a la condition $\Delta_{ik}(t) \geq \tau_{ik}$. On notera alors \mathcal{U}_t l'ensemble des tâches urgentes à la date t .

Sur la base tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{11}) .

3.3.2 Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{11})

Comme nous l'avons dit plus haut, pour la procédure temps-réel de résolution du problème (\mathcal{P}_{11}) , les décisions seront basées sur des problèmes localement statiques et la procédure de décision locale est basée sur un choix localement optimal pour le critère de coût en question. Pour le problème (\mathcal{P}_{11}) , nous allons donc baser la procédure sur la fonction PRTF, ses propriétés et les définitions et proposition ci-dessus.

Algorithme 3.3.2.1. Procédure en temps-réel OL-MTSA

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques sur chaque entité du système comme définies dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource).
La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les machines de tous les sites}\}$.
4. Déterminer dans l'ensemble \mathcal{S} l'entité à traiter en sélectionnant celle dont la tâche donne la plus petite valeur de la fonction PRTF pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} à sa date de fin.
Aller à l'étape 1.
5. Fin.

Lorsqu'il faut résoudre un problème statique d'ordonnancement de n tâches avec la fonction

PRTF, la complexité de la procédure utilisant cette fonction est en $O(n \cdot \log(n))$. Dans la procédure ci-dessus, la complexité est donnée dans la proposition ci-après.

Proposition 3.3.2.2. Complexité

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA est en $O(N' (\sum_{k=1}^N N_k))$. Cette complexité pouvait être en $O(N' \log(\sum_{k=1}^N N_k))$ sans la détermination de l'urgence des tâches.

Preuve. En effet, chaque décision prise l'est en sélectionnant dans un premier temps les entités qui ont des tâches urgentes. Cela se fait en $O(\sum_{k=1}^N N_k)$. Dans cet ensemble, l'entité fournissant la plus petite valeur de la fonction PRTF est déterminée. Cela se fait, en utilisant un arbre binaire, en $O(\log(\sum_{k=1}^N N_k))$. Ainsi, chaque décision est prise en $O(\sum_{k=1}^N N_k) + O(\log(\sum_{k=1}^N N_k))$, soit en $O((\sum_{k=1}^N N_k))$. Et comme cette décision est prise N' fois, on a alors une complexité en $O(N' (\sum_{k=1}^N N_k))$. On peut voir évidemment que sans l'étape de la détermination des tâches urgentes, on aurait une complexité en $O(N' \log(\sum_{k=1}^N N_k))$. \square

3.3.3 Protocole d'expérimentation

Les tests numériques sur les approches présentées dans cette section et dans celles qui vont suivre ont été effectués sur des données aléatoirement générées. Nous présentons ici un résumé des données qui vont être utilisées dans l'ensemble de ce chapitre. Les données représentant les taux de défaillance (λ_{ik}) et de réparation (μ_{ik}) des entités sont générées suivant des lois normales et sont prises comme étant des valeurs absolues de ces lois normales. Sur la base des taux de réparation, le temps moyen d'une tâche de maintenance sur chaque entité \mathcal{E}_{ik} (MTP_{ik}) dans le système est évalué comme étant son MTTR. Les seuils critiques de disponibilité (α_{ik}^1) sont générés suivant des lois uniformes et en respectant les disponibilités asymptotiques $\mu_{ik}/(\mu_{ik} + \lambda_{ik})$ des entités et sont toujours entre]0,5 ; 1[. Les durées τ_{ik} sont alors évaluées sur la base de la fonction disponibilité et de ces seuils de disponibilité.

L'objet de ce document étant un système distribué que nous reconsidérerons explicitement comme tel dans la suite, nous avons considéré un ensemble de sites sur lesquels la répartition des entités par site est uniforme discrète entre 20 et 80 entités par site (avec donc une moyenne de 50 entités par site). Par ailleurs, les systèmes amplement distribués impliquent d'importants temps logistiques entre les différents sites. Ces temps logistiques que nous reconsidérerons explicitement dans la suite sont pris suivant des valeurs absolues de lois normales. En outre, les poids des entités (w_{ik}) dépendant de la situation de celles-ci dans le système, et que nous reconsidérerons dans la suite comme tels, sont générés suivant des lois normales. Les figures 3.3 résument les distributions des données réelles obtenues dans les générations et telles qu'utilisées dans ce chapitre.

Les programmes sont toujours écrits en langage C et testés sur une station de travail Compaq AlphaServer ES40 DEC6600 opérant sous le système UNIX avec 2048 MB de mémoire RAM. Les tests sont tous entrepris sur un horizon \mathcal{H} de 365 jours (une année). Dans cette section comme dans les suivantes, les approches développées sont toujours programmées en deux versions. La première

version prend en compte la règle d'optimalité locale en addition avec le concept d'urgence des tâches de maintenance. Dans une deuxième version, seule la règle d'optimalité locale est utilisée pour la prise de décision en temps-réel dans le processus de maintenance. De cette façon, les paramètres représentant les résultats sont toujours suivis d'un suffixe : U si le concept d'urgence est utilisé dans la prise de décision et NU dans le cas contraire.

3.3.4 Résultats numériques

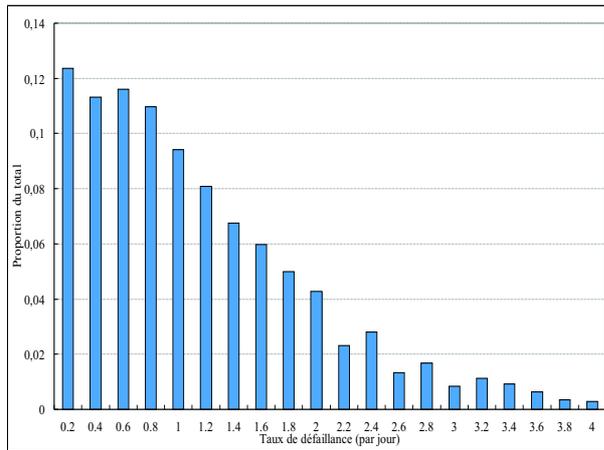
Expérimentations

Les résultats sont présentés sur un système comprenant 500 entités au total. Ces entités sont réparties sur les différents sites, mais comme pour le moment les temps logistiques entre les sites sont faibles, ces entités sont considérées toutes ensemble formant un bloc. Ensuite, comme la performance du processus de maintenance dépend du nombre de ressources chargées des tâches de maintenance, nous avons fait varier le nombre total de ressources pour tout le système en proportion du nombre total d'entités dans le système. Ce nombre de ressources varie entre 2% et 20% du nombre total d'entités variant donc entre 10 et 100 ressources. Les résultats ne sont pas représentés en fonction de ce nombre de ressources mais plutôt en fonction de la proportion du nombre total d'entités que représente ce nombre de ressources. La synthèse des résultats est présentée sur la figure 3.4.

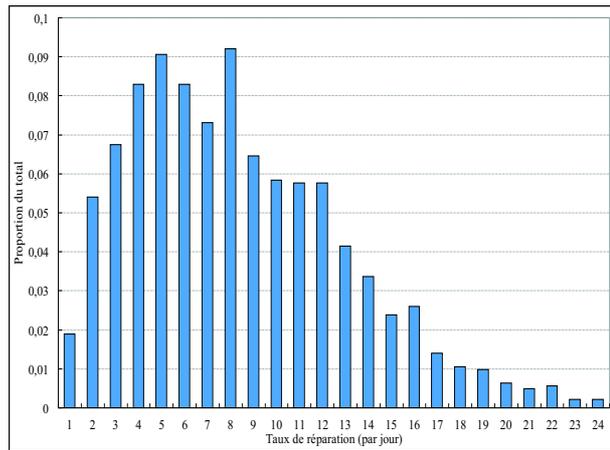
Cette figure 3.4 présente quatre type de paramètres. Sur la figure 3.4(a), le paramètre CMI représente les coûts moyens des états critiques sur le système sur l'horizon des 365 jours restreint aux tâches de maintenance qui ont effectivement été réalisées. Mais puisque ce coût est intimement lié au nombre de ressources de maintenance présentes, toutes les tâches requises de maintenance n'ont pas pu être réalisées sur l'horizon. Certaines sont repoussées à plusieurs reprises jusqu'à la fin de cet horizon. Celles qui n'ont pas pu être réalisées ont parfois des coûts très importants. C'est pourquoi le paramètre CMS qui est le coût moyen relevé sur l'ensemble du système (aussi bien sur les tâches réalisées que celle non réalisées) vient compléter le coût moyen CMI. Ces coûts d'états critiques représentent les temps de réponse moyens des tâches de maintenance sur l'horizon de planification \mathcal{H} . La figure 3.4(b) quant à elle représente le délai moyen pour les tâches de maintenance. Sur cette figure, le paramètre DMI est le correspondant de CMI en terme des délais alors que DMS est le correspondant de CMS. Sur la figure 3.4(c) se trouve le temps moyen d'activité par ressource (TMTpR) alors que la figure 3.4(d) elle représente le nombre moyen (NMIpR) de sollicitations de chacune des ressources sur l'horizon des 365 jours. Chacun de ces paramètres est relevé lorsque le concept d'urgence est utilisé dans la prise de décision (U), et (NU) lorsqu'il ne l'est pas.

Analyse des résultats

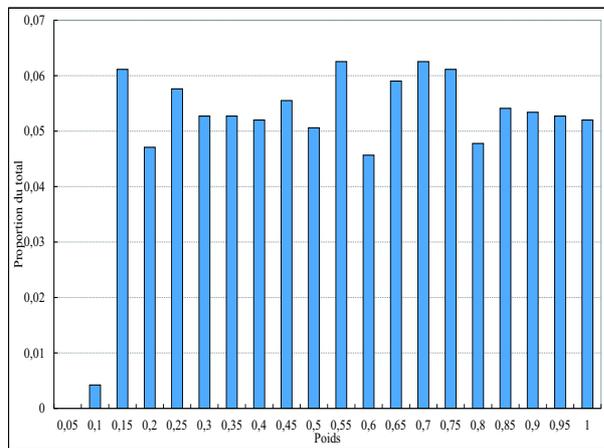
Pour des raisons de concision, nous n'allons pas faire une présentation exhaustive de l'analyse des résultats ici. Nous inviterons le lecteur à voir par exemple dans les papiers [8] et [10]. Néanmoins, il faut relever quelques observations.



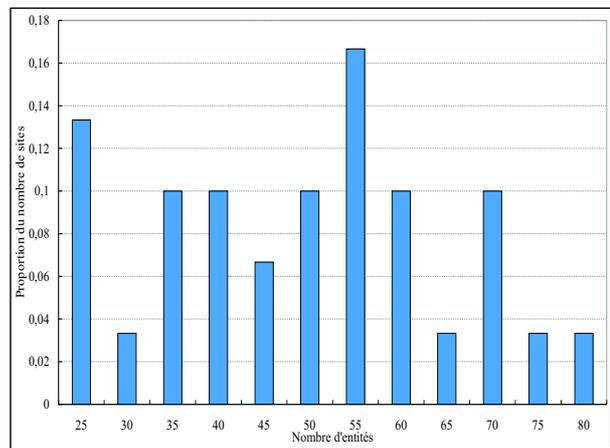
(a) Distribution des taux de défaillance



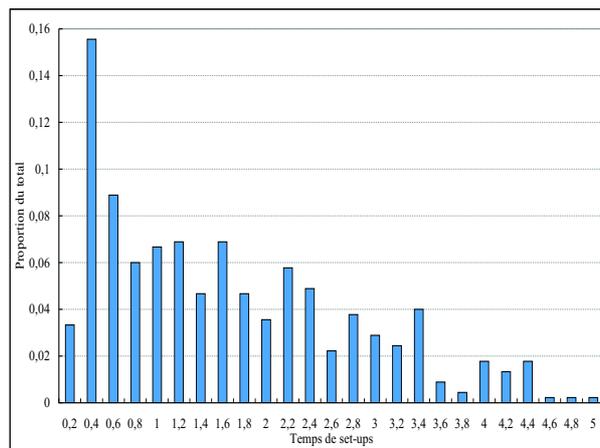
(b) Distribution des taux de réparation



(c) Distribution des poids

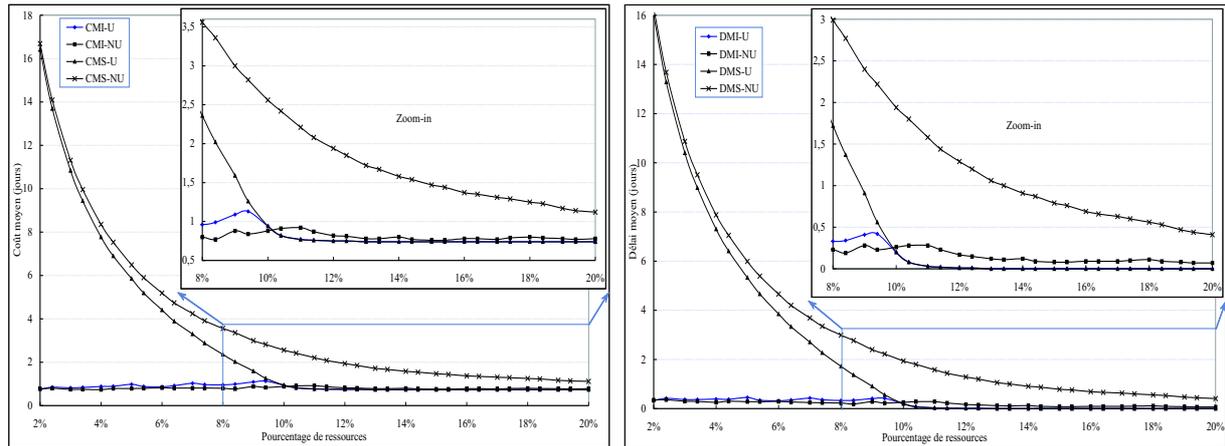


(d) Distribution du nombre d'entités par site



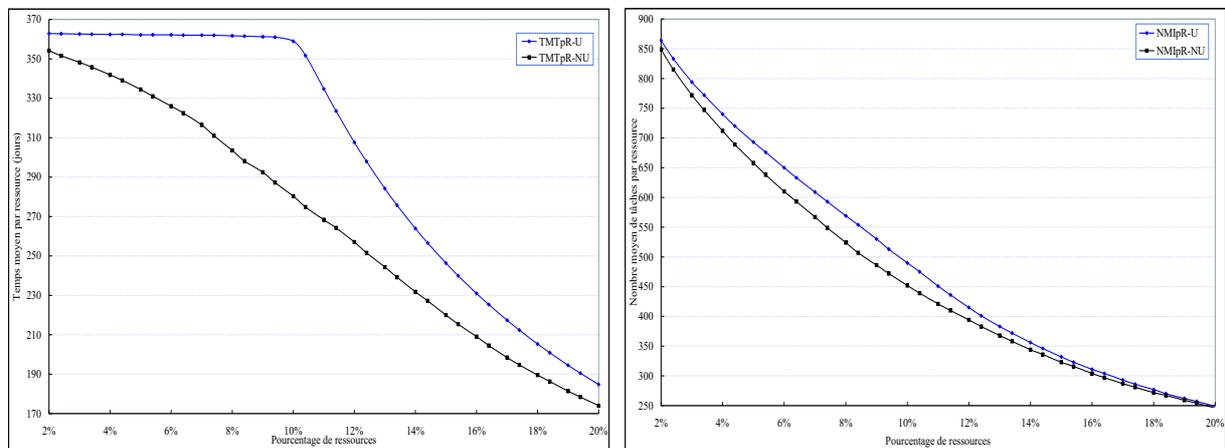
(e) Distribution des temps logistiques

Figure 3.3 – Distribution des données utilisées du système à maintenir



(a) Coût moyen de maintenance (en temps de réponse)

(b) Délai moyen d'intervention



(c) Temps moyen d'activité par ressource

(d) Nombre moyen de tâches par ressource

Figure 3.4 – Résultats avec des poids égaux et des temps logistiques faibles

Sur la figure 3.4(d), on voit le nombre considérable de tâches de maintenance réalisées sur le système des 500 entités sur l'horizon des 365 jours. Le nombre total est obtenu en multipliant le nombre de ressources par le nombre moyen de tâches de maintenance. Cela se fait en moins de 1 seconde CPU en général.

Sur la figure 3.4(a), on voit (pour un nombre faible de ressources de maintenance) la différence importante entre les coûts CMI et CMS (aussi bien avec le concept d'urgence que sans ce concept). On voit que lorsque le concept d'urgence est utilisé, ces deux coûts convergent et se confondent à partir de 10% de ressources. La convergence de ces deux coûts signifie que l'ensemble des sollicitations de tâches de maintenance sur l'horizon a trouvé une réponse. À partir de cette convergence, on observe sur la courbe 3.4(b) (où il y a aussi une très grande différence pour les nombres faibles de ressources entre les délais DMI et DMS) que ces deux délais convergent aussi avec l'utilisation du concept d'urgence et qu'après cette convergence, le délai (avec le concept d'urgence) s'annule

à partir de 11% de ressources. On voit sur les courbes 3.4(a) et 3.4(b) que si le concept d'urgence n'est pas utilisé, le coût sur l'ensemble du système est nettement plus grand que si le concept d'urgence est utilisé pour gérer les états critiques. Par ailleurs, on voit que si l'urgence n'est pas utilisée en plus de la relation d'optimalité locale, la convergence des deux coûts CMI et CMS et des deux délais DMI et DMS (même si cette convergence est possible), ne se réaliserait que largement au delà des 20% de ressources.

En outre, on voit sur la courbe 3.4(c) avec l'urgence que les ressources sont utilisées à leur pleine capacité lorsqu'il n'y a pas convergence entre CMI et CMS (et DMI et DMS) (c'est à dire que toutes les sollicitations sur le système ne sont pas couvertes). Le temps moyen d'utilisation ne commence à décroître rapidement qu'à partir du point 10% de convergence. Alors que sans l'urgence, le temps moyen d'utilisation par ressource décroît lorsque le pourcentage de ressources augmente (même lorsqu'il y a des tâches à entreprendre). Et ce temps d'utilisation est nettement inférieur à celui où l'urgence est utilisée conjointement avec la fonction d'optimalité locale.

Ainsi, en plus d'aider à maîtriser le coût des états critiques et des délais d'intervention, la procédure en temps-réel avec la fonction d'urgence et la fonction d'optimalité locale permet aussi de dimensionner le nombre de ressources nécessaires par rapport à la taille et aux caractéristiques du système.

Après cette première partie sur l'aspect le plus simple du problème (\mathcal{P}_1) avec des temps logistiques faibles entre les sites et des entités de même poids, nous allons maintenant considérer le problème (\mathcal{P}_1) dans sa deuxième configuration qui est (\mathcal{P}_{12}).

3.4 Configuration avec des entités pondérées et des temps logistiques faibles

Dans cette section, nous considérons le problème (\mathcal{P}_1) sous sa forme (\mathcal{P}_{12}). Dans cette configuration, les différentes entités de chaque site du système sont pondérées (et ont des poids inégaux) et les temps logistiques entre les sites sont aussi faibles comme dans la section précédente. Nous sommes ici avec un problème où les tâches sont donc pondérées, ont des dates d'occurrence inégales et où les temps logistiques sont faibles et assimilés à 0. Il faut donc minimiser le coût des états critiques

$$C_{tot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} w_{ik} (c_{m,ik} - r_{m,ik}) \right) \quad (3.4.1)$$

C'est un problème NP-difficile de prise de décision en temps-réel pour le processus de maintenance. Comme dans la section précédente, les prises de décisions dans ce problème (\mathcal{P}_{12}) se feront avec des choix locaux sur des problèmes localement statiques. Nous allons donc considérer dans un premier temps un ensemble de n tâches (disponibles ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Chaque tâche i a une date d'occurrence r_i (qui ne sont pas égales pour toutes les tâches), une durée d'exécution p_i , un poids w_i (qui diffère d'une tâche à une autre) et l'objectif est de minimiser leur temps total de réponse pondéré (ou temps total de séjour pondéré) qui est $\sum_{i=1}^n w_i (c_i - r_i)$ si la date de complétion de la tâche i est c_i .

Nous considérons en premier lieu le problème sur une ressource et cherchons les options pour une décision garantissant une optimalité locale.

3.4.1 Le problème statique et les règles de décision

En ce qui concerne la minimisation du temps de réponse pondéré avec des dates d'arrivée inégales des tâches, peu de travaux ont été publiés. Les procédures que nous décrivons dans cette section permettent non seulement une prise en compte du critère de poids, mais aussi une utilisation dans des algorithmes dynamiques, grâce à leur complexité et la façon dont elles sont établies et utilisées.

Soit un ensemble de n tâches indépendantes, chaque tâche i ayant sa date d'arrivée r_i , sa durée d'exécution p_i et son poids w_i . Le problème consiste à définir les conditions pour qu'une tâche i soit ordonnancée avant une autre tâche j , afin de minimiser le temps de réponse pondéré total $\sum_{i=1}^n w_i(c_i - r_i)$ de la séquence des tâches où c_i est la date de fin de la tâche i . Les tâches sont exécutées sans préemption. Nous allons dans un premier temps procéder à la recherche d'optimalité locale. Les conditions obtenues permettent d'établir une règle de priorité entre les tâches considérées deux par deux. Pour cela, nous allons utiliser la fonction ci-après définie. En fait, à cause des poids des tâches, nous aurons à définir certains concepts pour l'utilisation de la règle d'optimalité locale qui sera trouvée.

Définition 3.4.1.1. *Pour toute paire de tâches $\{i, j\}$, et à tout instant t , la fonction $PRTWF$ (Priority Rule for Total Weighted Flow-time) est définie comme suit :*

$$PRTWF(i, j, t) = (w_i + w_j) \max(r_i, t) + w_j p_i \quad (3.4.2)$$

Comme nous pouvons le noter, la fonction $PRTWF$ définie dans 3.4.1.1 considère une paire de tâche pour tous les calculs à chaque instant. Elle n'est pas fonction que d'une tâche à la fois. Mais le théorème 3.4.1.5 en fait une règle de priorité locale. Cette règle de priorité permettra de prendre des décisions localement optimales. Puisqu'elle ne dépend pas d'une tâche à la fois comme c'était le cas de la fonction d'optimalité locale $PRTF$ lorsque les tâches ne sont pas pondérées, nous introduisons la définition suivante qui sera utile pour son utilisation.

Définition 3.4.1.2. *Soit une paire de tâches $\{i, j\}$ à ordonnancer à l'instant t à la fin d'une séquence partielle π .*

i) Nous disons que i domine j à l'instant t et notons $i \preceq j$, si en ordonnancant i à la fin de π avant j , on obtient une plus petite valeur du coût total (ici le temps de réponse pondéré total) qu'en ordonnancant j avant i .

ii) Nous construisons une matrice notée Ω telle que:

$$\Omega(i, j) = \begin{cases} 1 & \text{si } i \neq j \text{ et } i \preceq j \\ 0 & \text{sinon.} \end{cases} \quad (3.4.3)$$

Remarque 3.4.1.3. *Dans le vocabulaire classique d'ordonnancement, le terme dominance se rapporte aux ordonnancements partiels et non aux tâches. Mais dans la définition 3.4.1.2, on peut remarquer que si à l'instant t , π est un ordonnancement partiel, dire que $i \preceq j$ équivaut à dire que la séquence $\pi i j$ domine $\pi j i$ dans le langage classique d'ordonnancement.*

Pour deux tâches i et j différentes donnant la même augmentation partielle du coût, nous utilisons leurs indices pour établir leur dominance. Ainsi, nous avons la remarque suivante :

Remarque 3.4.1.4. *Si la tâche i est différente de la tâche j , alors $\Omega(i,j) + \Omega(j,i) = 1$. Ainsi $\Omega(j,i)$ peut être écrit à partir de l'expression suivante :*

$$\Omega(j,i) = (1 - \Omega(i,j)) \cdot \mathbb{I}_{i \neq j}. \quad (3.4.4)$$

où \mathbb{I}_A est la fonction indicatrice de A .

Le calcul de la matrice Ω peut alors être réduit au calcul de la matrice triangulaire supérieure. Sur la base de la fonction $PRTWF$ et de la relation de dominance \preceq dans la définition 3.4.1.2, nous avons le théorème suivant :

Théorème 3.4.1.5. *À tout instant t et pour une paire de tâches $\{i,j\}$, $i \preceq j$ si et seulement si*

$$(w_i + w_j) \max(r_i, t) + w_j p_i \leq (w_i + w_j) \max(r_j, t) + w_i p_j \quad (3.4.5)$$

Autrement dit, $i \preceq j$ si et seulement si $PRTWF(i,j,t) \leq PRTWF(j,i,t)$.

Preuve. Les paramètres r_i, p_i, w_i sont affectés à la tâche i , et r_j, p_j, w_j à la tâche j . Supposons que $r_i \leq r_j$.

Il peut être vérifié que les conditions suivantes sont satisfaites en ordonnant les tâches i et j à l'instant t .

i) Si $r_i \leq r_j \leq t$, alors $i \preceq j$ si et seulement si $w_j p_i - w_i p_j \leq 0$.

ii-a) Si $r_i \leq t \leq r_j$ et $t + p_i \leq r_j$, alors $i \preceq j$ est évident.

ii-b) Si $r_i \leq t \leq r_j$ et $t + p_i > r_j$, alors $i \preceq j$ si et seulement si $[(w_i + w_j)t + w_j p_i] - [(w_i + w_j)r_j + w_i p_j] \leq 0$.

iii-a) Si $t \leq r_i \leq r_j$ et $r_i + p_i \leq r_j$, alors $i \preceq j$ est évident.

iii-b) Si $t \leq r_i \leq r_j$ et $r_i + p_i > r_j$, alors $i \preceq j$ si et seulement si $[(w_i + w_j)r_i + w_j p_i] - [(w_i + w_j)r_j + w_i p_j] \leq 0$.

Tous les cas i), ii-b) et iii-b) peuvent être résumés comme suit

$$i \preceq j \text{ si et seulement si } (w_i + w_j) \max(t, r_i) + w_j p_i \leq (w_i + w_j) \max(t, r_j) + w_i p_j.$$

Dans le cas ii-a), nous avons $r_i \leq t \leq r_j$ et $t + p_i \leq r_j$. A partir de la seconde de ces inégalités, nous avons

$$(w_i + w_j)(t + p_i) \leq (w_i + w_j)r_j$$

qui implique

$$(w_i + w_j)t + w_j p_i \leq (w_i + w_j)(t + p_i) \leq (w_i + w_j)r_j \leq (w_i + w_j)r_j + w_i p_j.$$

Le premier et le dernier membres de cette chaîne d'inégalités, en conjonction avec la première inégalité dans le cas ii-a) donnent

$$(w_i + w_j) \max(t, r_i) + w_j p_i \leq (w_i + w_j) \max(t, r_j) + w_i p_j.$$

Dans le cas iii-a), nous avons aussi $t \leq r_i \leq r_j$ et $r_i + p_i \leq r_j$. De la seconde inégalité, nous avons

$$(w_i + w_j)(r_i + p_i) \leq (w_i + w_j)r_j,$$

qui implique

$$(w_i + w_j)r_i + w_j p_i \leq (w_i + w_j)(r_i + p_i) \leq (w_i + w_j)r_j \leq (w_i + w_j)r_j + w_i p_j.$$

Le premier et le dernier membres de cette dernière chaîne d'inégalités, en plus de la première inégalité dans le cas iii-a) permettent de dire que

$$(w_i + w_j)\max(t, r_i) + w_j p_i \leq (w_i + w_j)\max(t, r_j) + w_i p_j.$$

Ceci prouve que dans tous les cas où $r_i \leq r_j$, l'on a

$$i \preceq j \text{ si et seulement si } (w_i + w_j)\max(t, r_i) + w_j p_i \leq (w_i + w_j)\max(t, r_j) + w_i p_j.$$

Supposons maintenant que $r_i > r_j$.

En inversant les rôles de i et j dans l'expression précédente, nous obtenons

$$j \preceq i \text{ si et seulement si } (w_i + w_j)\max(t, r_j) + w_i p_j \leq (w_i + w_j)\max(t, r_i) + w_j p_i$$

qui est la même chose que

$$i \preceq j \text{ si et seulement si } (w_i + w_j)\max(t, r_i) + w_j p_i \leq (w_i + w_j)\max(t, r_j) + w_i p_j.$$

Donc dans tous les cas, nous avons

$$i \preceq j \text{ si et seulement si } (w_i + w_j)\max(t, r_i) + w_j p_i \leq (w_i + w_j)\max(t, r_j) + w_i p_j.$$

Cela peut se réécrire, en utilisant l'expression 3.4.2 de la fonction $PRTWF$, comme

$$i \preceq j \text{ si et seulement si } PRTWF(i, j, t) \leq PRTWF(j, i, t). \quad \square$$

Il faut aussi faire la remarque ci-après.

Remarque 3.4.1.6. *La relation dominance \preceq dans la définition 3.4.1.2 basée sur la fonction $PRTWF$ ne définit pas une relation d'ordre total dans l'ensemble des tâches.*

Mais l'utilisation de la fonction $PRTWF$ pourra se faire efficacement, en utilisant les propriétés de la matrice de dominance Ω , sur la base de la définition qui suit.

Définition 3.4.1.7. *A tout instant t , la force f_i d'une tâche i est le nombre de tâches que i domine. En d'autres termes,*

$$f_i = \sum_j \Omega(i, j). \quad (3.4.6)$$

Le vecteur des forces satisfait à la propriété suivante qui permet de vérifier leur calcul dans une manipulation manuelle. Il faut noter que nous n'utiliserons pas cette propriété dans les programmes.

Proposition 3.4.1.8. *Si n est le nombre de tâches à ordonnancer à l'instant t , alors le vecteur des forces f_i vérifie l'égalité*

$$\sum_i f_i = \frac{n(n-1)}{2}. \quad (3.4.7)$$

Preuve. En effet,

$$\begin{aligned} \sum_i f_i &= \sum_i \sum_j \Omega(i, j) \\ &= \sum_{i, j, j \leq i} \Omega(i, j) + \sum_{i, j, j \geq i} \Omega(i, j), \text{ parce que } \Omega(i, i) = 0 \end{aligned}$$

c'est à dire

$$\begin{aligned}\sum_i f_i &= \sum_{\{i,j\}, j \geq i} (\Omega(i,j) + \Omega(j,i)) \\ &= \sum_{\{i,j\}} (\Omega(i,j) + \Omega(j,i)), \text{ à cause de la symétrie.}\end{aligned}$$

Or, $\Omega(i,j) + \Omega(j,i) = 1$ si $i \neq j$, et le nombre de telles combinaisons dans lesquelles $i \neq j$, est C_n^2 , qui est égale à $n(n-1)/2$. \square

Remarque 3.4.1.9. Dans le cas particulier où les tâches ont le même poids i.e. $w_i = 1$ ($i=1, \dots, n$), la fonction *PRTWF* devient

$$PRTWF(i,j,t) = 2\max(r_i, t) + p_i \quad (3.4.8)$$

Cette dernière expression dépend seulement d'une tâche à la fois et on remarquera qu'elle correspond à la fonction *PRTF* (Priority Rule for Total Flow-time), règle de priorité localement optimale dans la définition 3.3.1.1 pour la minimisation du temps réponse dans le cas où les tâches ne sont pas pondérées mais ont des dates d'occurrence inégales.

Les deux définitions de dominance et de forces des tâches vont être utilisées dans le cas présent, mais aussi seront très importantes dans la suite de ce travail. Elles permettront une utilisation efficace de règles qui, à première vue, ne sont pas évidentes d'utilisation.

3.4.2 Situation de la fonction *PRTWF* par rapport aux règles de dominance de la littérature

Avant de passer au principe d'utilisation de la fonction *PRTWF*, de la matrice de dominance Ω et du vecteur de force des tâches, nous allons situer cette règle par rapport à des propriétés de dominance utilisées dans la littérature pour la minimisation du temps de réponse pondéré en présence de dates d'arrivée inégales des tâches. Il faut noter que la règle fournie par la fonction *PRTWF* est une condition nécessaire et suffisante de dominance. Les propriétés de la littérature que nous allons citer ci-après sont des conditions suffisantes. Ces propriétés se retrouvent dans [24] et [131]. Nous les citons en utilisant les notations de ce document-ci.

Théorème 3.4.2.1. Soit un ensemble de n tâches, une séquence partielle π_k de tâches terminée par la tâche k ($k < n$), où t est la date de complétion pour π_k , et une tâche i pas encore ordonnancée. Si toutes les tâches $j \neq i$ non encore ordonnancées vérifient les deux conditions suivantes par rapport à la tâche i ,

$$(a) \frac{p_i}{w_i} \leq \frac{p_j}{w_j}$$

$$(b) \max(r_i, t) \leq \max(r_j, t)$$

alors i domine j .

Théorème 3.4.2.2. Soit un ensemble de n tâches, une séquence partielle π_k ($k < n$) où t est la date de complétion pour π_k , et deux tâches i, j non encore ordonnancées,

$$\text{si } r_j \geq \max(r_i, t) + p_i, \text{ alors } i \text{ domine } j.$$

Théorème 3.4.2.3. *Soit un ensemble de n tâches, une séquence partielle π_k ($k < n$) ayant comme date de complétion t et deux tâches i et j non encore ordonnancées ; si*

$$(a) w_i \geq w_j$$

$$(b) \max(r_j, t) + p_j \geq \max(r_i, t) + p_i$$

$$(c) w_i[\max(r_j, t) + p_j - \max(r_i, t) - p_i] + w_i p_i - w_j p_j \geq (p_j - p_i) \sum_{l \in Q} w_l$$

où Q est l'ensemble des tâches non encore ordonnancées privé des tâches i et j , alors i domine j .

Théorème 3.4.2.4. *Soit un ensemble de n tâches, une séquence partielle π_k ($k < n$) de date de complétion t et deux tâches i et j non encore ordonnancées ; si*

$$(a) w_i \geq w_j$$

$$(b) \max(r_j, t) + p_j \leq \max(r_i, t) + p_i$$

$$(c) w_i p_i - w_j p_j \geq [\max(r_i, t) + p_i - \max(r_j, t) - p_j] \sum_l w_l + \delta_{-1}(p_j - p_i) \cdot [\sum_l w_l - w_j - w_i]$$

où δ est le symbole de Kronecker et la somme est effectuée sur l'ensemble des tâches non encore ordonnancées (noté \bar{K}) ; alors i domine j .

Théorème 3.4.2.5. *Soit un ensemble de n tâches, une séquence partielle π_k ($k < n$) de date de complétion t et deux tâches i et j non encore ordonnancées. Notons \bar{K} (ensemble de tâches non encore ordonnancées) l'ensemble "dense" (dans le sens définis par Bianco et Ricciardelli dans [24]), i.e. l'ensemble dans lequel il n'y a pas de temps de latence. Si*

$$(a) w_i \geq w_j$$

$$(b) \max(r_j, t) \leq \max(r_i, t)$$

$$(c) \max(r_j, t) + p_j \leq \max(r_i, t) + p_i$$

alors i domine j .

Les théorèmes 3.4.2.1 à 3.4.2.5 décrivent des conditions suffisantes de dominance utilisées dans [24] et [131] pour traiter des instances de tâches allant jusqu'à 10 tâches dans [24]. Mais dans un processus de décision en temps-réel ou dans un processus de décision basé sur des critères d'optimalité locale, les conditions décrites dans l'ensemble de ces théorèmes deviennent très lourdes à gérer et, en plus, ils peuvent tous ne pas être applicables à un instant donné lorsque les hypothèses ne sont pas vérifiées.

Remarque 3.4.2.6. *Il peut être vérifié facilement que les hypothèses de chacun des théorèmes 3.4.2.1 à 3.4.2.5 se ramènent au théorème 3.4.1.5 avec la fonction PRTWF.*

Nous allons maintenant donner le principe de base pour l'utilisation de la matrice de dominance, du vecteur des forces des tâches et de la fonction PRTWF. Pour cela, nous considérons un ensemble de n tâches pondérées avec des dates d'occurrence inégales à ordonnancer sur une ressource afin de minimiser leur temps total de réponse pondéré. La procédure considérée ici est une procédure en statique où l'on connaît toutes les caractéristiques de toutes les tâches (présentes et à venir) avant même le début de l'ordonnancement.

3.4.3 Algorithme de base

Algorithme 3.4.3.1. Principe de base du PRTWF : AlgoOF-PRTWF

Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée, les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$
 2. Calculer $PRTWF(i,j,t)$ pour tous les couples de tâches (i,j) dans S .
 3. Calculer la matrice de dominance Ω dans S .
 4. Calculer le vecteur des forces (f_i) dans S .
 5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
- Tant que $\text{card}(S) > 1$, aller à l'étape 4.*

Il faut noter qu'à chaque prise de décision, la matrice donnant les valeurs de la fonction PRTWF et la matrice de dominance ne sont calculées qu'une fois. Lorsqu'on réduit l'ensemble de recherche de la tâche à ordonnancer, la date de décision ne change pas, ce qui fait que les valeurs dans ces deux matrices ne changent pas. Par contre, leur restriction au nouvel ensemble de recherche est considérée. Seul le vecteur des forces des tâches change, en ne tenant compte que des tâches dans ce nouvel ensemble de recherche.

Remarque 3.4.3.2. *Cet algorithme peut aussi être adapté dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égale à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant le plus petit indice parmi elles. La sélection des tâches se fait suivant les mêmes étapes que précédemment.*

La complexité de l'algorithme 3.4.3.1 est donnée dans la proposition ci-dessous.

Proposition 3.4.3.3. *L'algorithme AlgoOF-PRTWF basé sur le principe de la fonction PRTWF, de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$ si n est le nombre de tâches à ordonnancer.*

Preuve. Dans la procédure 3.4.3.1 (AlgoOF-PRTWF), les étapes 1 à 5 s'exécutent en $O(n^2)$. Ainsi, chaque décision est prise en $O(n^2)$. Ces étapes sont exécutées jusqu'à la dernière tâche. Cela se fait n fois. D'où la complexité de la procédure AlgoOF-PRTWF est en $O(n^3)$. \square

Il faut noter dans la procédure d'utilisation de la fonction PRTWF, la recherche de la tâche à ordonnancer est étendue à l'ensemble de toutes les tâches. Dans cette procédure, il peut être préférable à un instant de décision d'attendre pour exécuter une tâche à venir alors qu'il y a d'autres tâches en attente. Par ailleurs, dans les mêmes conditions que dans l'algorithme 3.4.3.1, une borne inférieure au temps de réponse pondéré total est donnée dans la proposition suivante. Elle considère

la préemption des tâches (une tâche commencée peut être interrompue au profit d'une autre) et le principe qui en résulte est appelé *SWRPT* (Shortest Weighted Remaining Processing Time). Dans cette proposition, $res(p_i)$ est la durée d'exécution restante de la tâche i à la date de décision t .

Proposition 3.4.3.4. *Lorsque la préemption des tâches est permise, l'ordonnancement dans lequel à tout instant t , la tâche i ayant le plus petit rapport $res(p_i)/w_i$ est ordonnancée, donne une borne inférieure au temps de réponse pondéré total dans la solution en statique.*

En d'autres termes, la tâche sélectionnée est celle dont l'indice est le plus petit parmi celles ayant ce plus petit rapport. A tout instant t , la tâche en exécution est interrompue s'il arrive une autre tâche ayant un plus petit rapport à la date t .

Par ailleurs, Hariri et Potts [75] ont développé une borne inférieure à ce problème en utilisant la relaxation lagrangienne des dates d'occurrences des tâches alors que Beloudah *et al.* [21] ont plutôt utilisé le splitting des tâches (avec préemption des tâches et décomposition de leurs poids). Il faut aussi noter que Hariri et Potts [75] et Beloudah *et al.* [21] ont proposé des méthodes de *branch and bound* qui résolvent les problèmes statiques pour des instances allant jusqu'à 50 tâches. Une solution de départ pour la méthode de Beloudah *et al.* se base sur une heuristique dans laquelle à un instant de décision t donné, seul l'ensemble des tâches dont la date d'occurrence r_i est inférieure ou égale à t est considéré. Dans cet ensemble de tâches, la tâche ayant le plus petit rapport p_i/w_i (durée opératoire sur poids) est ordonnancée. Dans cette procédure, à chaque décision, seules les tâches présentes sont considérées. Ainsi, l'insertion de temps mort entre les tâches est réduit.

Avant la procédure utilisant tout ce qui précède pour la résolution du problème \mathcal{P}_{12} , voyons avec des exemples simples comment s'utilisent ces différents éléments dans la pratique. Nous allons pour cela considérer deux instances de 5 tâches et 10 tâches respectivement (qui sont tirées d'un lot d'instances aléatoirement générées).

3.4.4 Exemples illustratifs de la procédure AlgoOF-PRTWF

Exemple 3.1

Considérons un ensemble de 5 tâches caractérisées par leur date d'occurrence (date d'arrivée) r_i , leur durée d'exécution p_i et leur poids w_i comme dans le tableau 3.1.

i	1	2	3	4	5
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03
Poids (w_i)	0,91	1,46	2,57	2,02	1,84

Table 3.1 – Instance de l'exemple 3.1 pour la fonction PRTWF

À l'instant $t = 0$, il faut évaluer l'ensemble de 5 tâches. Les valeurs de la fonction PRTWF, la matrice de dominance et le vecteur des forces des tâches à cette date sont représentés respectivement dans les tableaux 3.2, 3.3 et 3.4. Ici, on peut vérifier la proposition 3.4.1.8, en voyant que la

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	3,8114	5,9204	4,8754	4,5334
2	8,6952	-	17,7920	14,7780	13,7916
3	11,6146	13,5396	-	15,4996	14,8696
4	3,9215	4,8510	6,7269	-	5,4932
5	2,6173	3,8878	6,4519	5,1814	-

Table 3.2 – Valeurs de la fonction PRTWF à $t = 0$ dans l'exemple 3.1

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	0	0
2	0	-	0	0	0
3	0	1	-	0	0
4	1	1	1	-	0
5	1	1	1	1	-

Table 3.3 – Matrice de dominance à $t = 0$ dans l'exemple 3.1

Tâche i	1	2	3	4	5
Force f_i	2	0	1	3	4

Table 3.4 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.1

somme des forces des tâches est $\sum_i f_i = \frac{5(5-1)}{2} = 10$. Sur la base du vecteur des forces f_i , la tâche 5 est sélectionnée à la date $t = 0$.

Les autres instants de décision et les décisions à ces instants sont regroupés dans le tableau 3.5.

Instant de décision (t)	Ensemble des tâches	Tâche sélectionnée
2,31	{1, 2, 3, 4}	4
2,82	{1, 2, 3}	3
3,50	{1, 2}	1
4,26	{2}	2

Table 3.5 – Autres décisions dans l'exemple 3.1 pour la fonction PRTWF

Ordre d'exécution	1	2	3	4	5
Tâche	5	4	3	1	2
Date de début	0,28	2,31	3,28	3,50	4,26
Date de fin	2,31	2,82	3,50	4,26	7,20

Table 3.6 – Séquence d'exécution des tâches de l'exemple 3.1

La séquence d'exécution des tâches est donc représentée dans le tableau 3.6 et le coût moyen (temps de réponse pondéré moyen par tâche) est 3,3554. La borne inférieure à ce coût moyen fournie par règle SWRPT dans la proposition 3.4.3.4 est 2,9644. La valeur fournie par la procédure restreignant le choix aux seules tâches présentes est 3,4512.

Exemple 3.2

L'instance du deuxième exemple est la suite de celle de l'exemple 3.1 et comporte 10 tâches. Cette instance est dans le tableau 3.7.

i	1	2	3	4	5	6	7	8	9	10
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28	0,84	0,88	0,47	1,95	0,30
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03	2,10	2,10	0,10	0,58	0,31
Poids (w_i)	0,91	1,46	2,57	2,02	1,84	1,12	2,98	1,60	0,93	1,98

Table 3.7 – Instance de l'exemple 3.2 pour la fonction $PRTWF$

En procédant de la même façon que dans l'exemple 3.1, on retrouve la séquence des tâches dans le tableau 3.8 avec un temps de réponse pondéré moyen par tâche égale à 5,603. La borne inférieure à ce temps de réponse pondéré moyen par tâche fournie par la proposition 3.4.3.4 est 5,162.

Ordre d'exécution	1	2	3	4	5	6	7	8	9	10
Tâche	10	8	4	7	3	9	1	5	6	2
Date de début	0,30	0,61	1,18	1,69	3,79	4,01	4,59	5,35	7,38	9,48
Date de fin	0,61	0,71	1,69	3,79	4,01	4,59	5,35	7,38	9,48	12,42

Table 3.8 – Séquence d'exécution des tâches de l'exemple 3.2

Sur la base de tout ce qui précède, nous reconsidérons maintenant le problème (\mathcal{P}_{12}) dans sa forme temps-réel.

3.4.5 Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{12})

Dans la résolution du problème en temps-réel, la proposition suivante nous permettra dans la suite de cette section de parler indifféremment, tout comme dans le problème (\mathcal{P}_{11}), des entités que des demandes de tâches de maintenance sur ces entités. La justification de cette proposition est analogue à celle de la proposition 3.3.1.3.

Proposition 3.4.5.1. *En respectant le théorème 3.4.1.5 avec la fonction $PRTWF$, on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} . En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, le théorème 3.4.1.5 permet de programmer la $m_1^{ième}$ tâche sur \mathcal{E}_{ik} avant la $m_2^{ième}$.*

Ainsi, dans la suite de cette section, nous parlerons indifféremment de sélection d'une tâche sur une entité que de sélection d'une entité. Dans la procédure temps-réel, nous utiliserons en plus de la fonction $PRTWF$, du théorème 3.4.1.5 et de la proposition 3.4.5.1, le concept de tâches urgentes introduit dans la définition 3.3.1.4.

Sur tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{12}).

Pour la procédure temps-réel de résolution du problème (\mathcal{P}_{12}), les décisions seront basées sur des problèmes localement statiques et la procédure de décision locale est basée sur un choix localement optimal pour le critère de coût en question. Nous allons donc pour le problème (\mathcal{P}_{12}) baser la procédure sur la fonction PWRTF, ses propriétés et les caractéristiques de son utilisation pour les choix localement optimaux. Nous allons, outre la proposition 3.4.5.1 justifiant l'utilisation de la fonction PRTWF sur les entités, faire usage aussi de la notion d'urgence des tâches de maintenance comme introduite dans la définition 3.3.1.4. La procédure de décision en temps-réel pour cet aspect du problème (\mathcal{P}_1), avec des entités pondérées et des temps logistiques faibles entre les sites est appelé *OL-MTSA-W* dans l'algorithme 3.4.5.2.

Algorithme 3.4.5.2. Procédure temps-réel OL-MTSA-W

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques sur chaque entité du système comme définies dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-après.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer la procédure décrite dans l'algorithme 3.4.3.1 (AlgoOF-PRTWF) à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

La complexité de cette procédure en temps-réel est donnée dans la proposition suivante et est fonction du nombre total de tâches de maintenance entreprises sur l'horizon de planification \mathcal{H} .

Proposition 3.4.5.3. Complexité

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-W est en $O\left(N' \left(\sum_{k=1}^N N_k\right)^2\right)$.

Preuve. Pour chaque décision, il faut d'abord déterminer les entités ayant des tâches urgentes. Cela se fait en $O\left(\sum_{k=1}^N N_k\right)$. Puis l'entité avec la tâche la plus prioritaire, sur la base de la procédure 3.4.3.1 (AlgoOF-PRTWF). La sélection est faite, de façon analogue à AlgoOF-PRTWF, en $O\left(\left(\sum_{k=1}^N N_k\right)^2\right)$. Ainsi, chaque prise de décision est faite $O\left(\sum_{k=1}^N N_k\right) + O\left(\sum_{k=1}^N N_k\right)^2$, soit

en $O\left(\left(\sum_{k=1}^N N_k\right)^2\right)$. Et comme cela se fait N' fois sur l'horizon, on a donc une complexité en $O\left(N' \left(\sum_{k=1}^N N_k\right)^2\right)$. \square

3.4.6 Résultats numériques

Expérimentations

Pour les expérimentations, nous reconsidérons le protocole d'expérimentation de la sous-section 3.3.3 en considérant à présent les données relatives aux poids des entités. Comme dans la section précédente, les résultats sont présentés sur un système comprenant 500 entités au total. Ces entités sont réparties sur les différents sites mais comme les temps logistiques entre les sites sont faibles, ces entités sont considérées toutes ensemble formant un bloc. Ici aussi, nous avons fait varier le nombre total de ressources de maintenance pour tout le système en proportion du nombre total d'entités dans le système. Ce nombre de ressources varie entre 2% et 20% du nombre total d'entités variant donc entre 10 et 100 ressources. L'exécution des programmes se fait sur un horizon de 365 jours. Les résultats sont représentés en fonction de la proportion du nombre total d'entités que représente le nombre de ressources de maintenance. La synthèse des résultats est présentée sur la figure 3.5. Sur cette figure, nous gardons les mêmes appellations des paramètres que dans la section précédente. Mais ici, les coûts des états critiques CMI et CMS représentent les temps de réponse pondérés moyens de tâches de maintenance. Chacun de ces paramètres est relevé lorsque le concept d'urgence est utilisé dans la prise de décision (U), et (NU) lorsqu'il ne l'est pas.

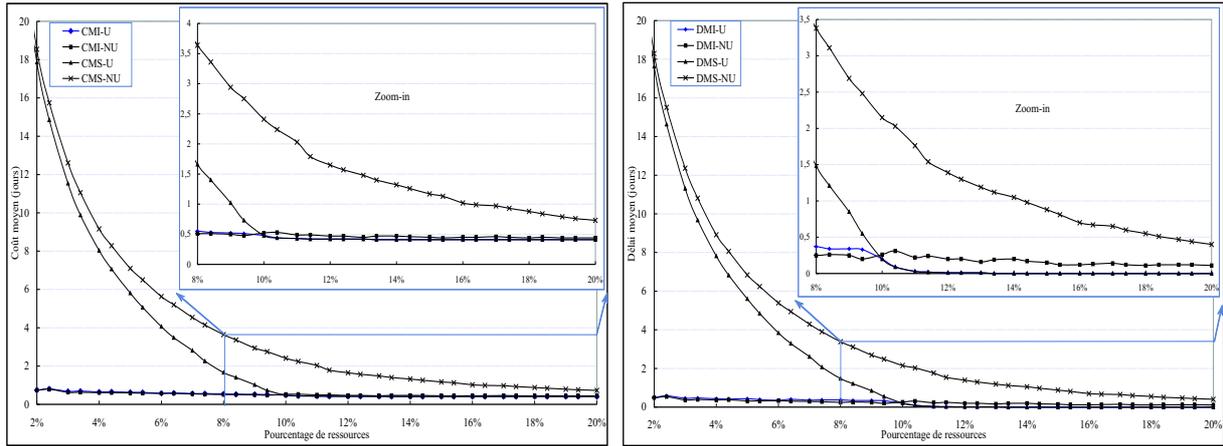
Analyse des résultats

Globalement, les observations sont analogues à celles de la sous-section 3.3.4. Pour davantage de détails sur l'analyse des résultats, le lecteur peut consulter par exemple [3] ou [9].

Nous allons maintenant considérer le problème (\mathcal{P}_1) dans sa troisième configuration (\mathcal{P}_{13}) avec des temps logistiques non négligeable entre les sites et des entités non pondérées (ou encore de mêmes poids égaux à 1).

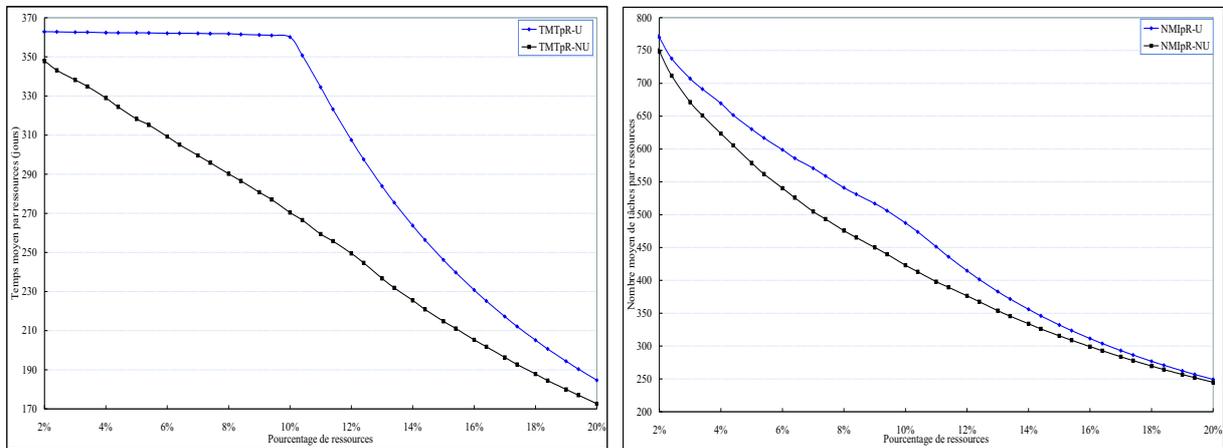
3.5 Configuration avec des entités non pondérées et des temps logistiques considérables

Cette section sera consacrée au problème (\mathcal{P}_1) sous sa forme (\mathcal{P}_{13}). Dans cette configuration, les différentes entités de chaque site du système sont non pondérées (ou de façon équivalente ont des poids tous égaux à 1) mais les temps logistiques entre les sites sont considérables de sorte que l'on ne puisse faire d'hypothèses réductrices sur eux à ce niveau. Nous sommes donc ici avec un problème où les tâches sont non pondérées, ont des dates d'occurrence inégales et où les temps logistiques sont importants entre les sites. En outre, ces temps logistiques entre les sites peuvent être symétriques ou non en ce sens que la configuration du système peut être telle que le temps



(a) Coût moyen de maintenance (en temps de réponse pondéré)

(b) Délai moyen d'intervention



(c) Temps moyen d'activité par ressource

(d) Nombre moyen de tâches par ressource

Figure 3.5 – Résultats avec des poids inégaux et des temps logistiques faibles

logistique d'un site S_k à un autre site S_l (s_{kl}) soit différent du temps logistique inverse s_{lk} . Il faut donc minimiser le coût des états critiques

$$C_{tot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} (c_{m,ik} - r_{m,ik}) \right) \quad (3.5.1)$$

en présence des temps logistiques importants entre les sites. Comme dans les précédentes configurations, c'est un problème NP-difficile de prise de décision en temps-réel pour le processus de maintenance. Comme précédemment, les prises de décisions dans ce problème (\mathcal{P}_{13}) se feront avec des choix locaux sur des problèmes localement statiques. Nous allons donc considérer dans un premier temps un ensemble de n tâches (disponibles ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Dans la phase de résolution du problème statique, nous assimilons les temps logistiques entre les sites aux temps logistiques entre des tâches devant être exécutées sur ces sites. Nous parlerons donc dans cette partie de temps logistiques entre les

tâches plutôt que de sites. Les temps logistiques entre les tâches dépendent de la séquence d'exécution celles-ci. Ils peuvent être symétriques ou non. Nous considérons au premier abord le problème sur une ressource et cherchons les options pour une décision garantissant une optimalité locale.

3.5.1 Le problème statique et les règles de décision

La résolution des problèmes d'ordonnement de tâches pour la minimisation du temps de réponse en présence de dates d'occurrence inégales des tâches et de temps logistiques dépendant de la séquence des tâches est quasiment introuvable dans la littérature.

Soit un ensemble de tâches indépendantes, chaque tâche i ayant sa date d'occurrence r_i , sa durée d'exécution p_i et un temps logistique s_{ij} vers la tâche j suivante. Le temps logistique s_{ij} est la durée des travaux préparatoires séparant la fin de la tâche i et le début de la tâche j . Dans le cadre d'un système distribué, ce temps logistique tient compte aussi du temps dont a besoin une ressource pour aller du site de la tâche i à celui de la tâche j . Si une tâche i est exécutée en première position, alors elle est précédée d'un temps logistique s_{0i} où la tâche 0 est une tâche fictive dont la date de complétion est apparentée à la date initial de début de l'ordonnement. Notons que le temps logistique s_{ij} dépend de la séquence des tâches ij et qu'à la fin d'une tâche i , le temps logistique s_{ij} vers une autre tâche j peut être exécutée avant même la date d'occurrence de cette dernière tâche.

Le problème ici consiste à établir les conditions pour qu'une tâche i soit ordonnancée avant la tâche j , afin de minimiser leur temps de réponse total (ou temps total de séjour) non pondéré $\sum_{i=1}^n (c_i - r_i)$ si la date de complétion de la tâche i est c_i . Ce problème NP-difficile et l'exigence de prise de décision en temps-réel dans la gestion des tâches de maintenance amène à la recherche de conditions d'optimalité locale. Comme dans le cas de la section précédente, les critères d'optimalité locale ne dépendent pas seulement d'une tâche à la fois. Mais nous pourrions les utiliser avec l'aide de quelques uns des concepts définis dans la section précédente. Nous allons pour cela réutiliser la notion de dominance d'une tâche sur une autre et la notion de force des tâches.

Pour commencer, nous avons dans le cas de la minimisation du temps de réponse total en présence de dates d'occurrence inégales des tâches et de temps logistiques entre ces tâches, le théorème ci-après. Dans ce théorème, nous supposons qu'il faut déterminer la tâche à ordonnancer à la fin d'une séquence partielle π qui se termine par la tâche k .

Théorème 3.5.1.1. *À tout instant t , si k est la dernière tâche dans la séquence partielle π , exécutée par la ressource, alors pour toute paire de tâches $\{i, j\}$, $i \preceq j$ si et seulement si*

$$[2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0. \quad (3.5.2)$$

Preuve. Soient i et j deux tâches à exécuter à la fin d'une séquence π terminée par la tâche k . La tâche i a les paramètres r_i , p_i et un temps logistique s_{ki} à partir de la fin de la tâche k au début de la tâche i ; la tâche j a les paramètres r_j , p_j et un temps logistique s_{kj} à partir de la fin de la tâche k au début de la tâche j . Le temps logistique de la tâche i à la tâche j est s_{ij} et celui de la tâche j à la tâche i est s_{ji} .

Supposons que $r_i \leq r_j$. Alors il peut être vérifié en ordonnant les tâches i et j à l'instant t à la fin de la tâche k , que les assertions suivantes sont satisfaites.

i) Si $r_i \leq r_j \leq t$ alors

$$i \preceq j \text{ si et seulement si } [2s_{ki} + p_i + s_{ij}] - [2s_{kj} + p_j + s_{ji}] \leq 0.$$

ii-a) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

ii-b1) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$$i \preceq j \text{ si et seulement si } [2t + 2s_{ki} + p_i + s_{ij}] - [2r_j + p_j + s_{ji}] \leq 0.$$

ii-b2) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$$i \preceq j \text{ si et seulement si } [2s_{ki} + p_i + s_{ij}] - [2s_{kj} + p_j + s_{ji}] \leq 0.$$

iii-a1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

iii-a2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

iii-b1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$$i \preceq j \text{ si et seulement si } [2r_i + p_i + s_{ij}] - [2r_j + p_j + s_{ji}] \leq 0$$

iii-b2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$$i \preceq j \text{ si et seulement si } [2r_i + p_i + s_{ij}] - [2t - 2s_{kj} + p_j + s_{ji}] \leq 0$$

iii-c1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$$i \preceq j \text{ si et seulement si } [2t + 2s_{ki} + p_i + s_{ij}] - [2r_j + p_j + s_{ji}] \leq 0$$

iii-c2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$$i \preceq j \text{ si et seulement si } [2s_{ki} + p_i + s_{ij}] - [2s_{kj} + p_j + s_{ji}] \leq 0.$$

Tous les cas i), ii-b1) ii-b2), iii-b1), iii-b2), iii-c1) et iii-c2) peuvent être résumés, en utilisant les différentes conditions et inégalités dans ces cas, comme suit :

$$i \preceq j \text{ si et seulement si } [2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

Dans le cas ii-a), nous avons : $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$.

La seconde inégalité est la même chose que

$$2(t + s_{ki}) + 2(p_i + s_{ij}) \leq 2r_j$$

qui implique

$$2(t + s_{ki}) + p_i + s_{ij} \leq 2(t + s_{ki}) + 2(p_i + s_{ij}) \leq 2r_j \leq 2r_j + p_j + s_{ji}.$$

Les deux membres extrêmes de cette chaîne d'inégalités donnent

$$2(t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji}.$$

En utilisant la première inégalité dans le cas ii-a), qui donne $r_i \leq t + s_{ki}$, nous avons à partir de l'inégalité précédente

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji}.$$

Cette dernière inégalité donne

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_i + s_{ji} \leq 2\max(r_j, t + s_{kj}) + p_i + s_{ij}.$$

On a alors, avec le premier et le dernier membre de cette chaîne d'inégalités,

$$[2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

Dans le cas iii-a1), nous avons :

$$t \leq r_i \leq r_j \text{ et } t + s_{ki} \leq r_i \text{ et } r_i + p_i + s_{ij} \leq r_j.$$

La troisième inégalité est la même chose que

$$2r_i + 2p_i + 2s_{ij} \leq 2r_j.$$

Cela donne

$$2r_i + p_i + s_{ij} \leq 2r_i + 2p_i + 2s_{ij} \leq 2r_j \leq 2r_j + p_j + s_{ji}.$$

A partir du premier et du dernier membre de cette chaîne, l'on a

$$2r_i + p_i + s_{ij} \leq 2r_j + p_j + s_{ji}.$$

Cela revient, en la combinant avec la seconde inégalité dans le cas iii-a1), à

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji},$$

ce qui donne enfin

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji} \leq 2\max(r_j, t + s_{kj}) + p_j + s_{ji}.$$

Les membres extrêmes de chaîne permettent de dire que

$$[2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

Dans le cas iii-a2), nous avons :

$$t \leq r_i \leq r_j \text{ et } t + s_{ki} > r_i \text{ et } t + s_{ki} + p_i + s_{ij} \leq r_j.$$

La troisième inégalité est équivalent à

$$2(t + s_{ki}) + 2p_i + 2s_{ij} \leq 2r_j,$$

qui donne

$$2(t + s_{ki}) + p_i + s_{ij} \leq 2(t + s_{ki}) + 2p_i + 2s_{ij} \leq 2r_j \leq 2r_j + p_j + s_{ji}.$$

Avec les premier et dernier membres de cette chaîne, nous avons

$$2(t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji}.$$

En combinant cette inégalité avec la seconde inégalité dans le cas iii-a2), nous avons

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji}.$$

qui permet de dire que

$$2\max(r_i, t + s_{ki}) + p_i + s_{ij} \leq 2r_j + p_j + s_{ji} \leq 2\max(r_j, t + s_{kj}) + p_j + s_{ji}.$$

Finalement, nous avons

$$[2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

Ainsi, dans tous les cas où $r_i \leq r_j$, on a

$$i \preceq j \text{ si et seulement si } [2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

Supposons maintenant que $r_i > r_j$. Alors, en inversant les rôles de i et j dans l'expression précédente, nous avons

$$j \preceq i \text{ si et seulement si } [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] - [2\max(r_i, t + s_{ki}) + p_i + s_{ij}] \leq 0.$$

qui revient à

$$i \preceq j \text{ si et seulement si } [2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0.$$

En conclusion, nous avons dans tous les cas

$$i \preceq j \text{ si et seulement si } [2\max(r_i, t + s_{ki}) + p_i + s_{ij}] - [2\max(r_j, t + s_{kj}) + p_j + s_{ji}] \leq 0. \quad \square$$

Nous posons ainsi les définitions suivantes.

Définition 3.5.1.2. Pour toute paire de tâches $\{i, j\}$ et à tout instant t où une tâche k est complétée à la fin d'une séquence partielle π , la fonction *PRTFST* (Priority Rule for Total Flow-time with Set-up Times) est définie par :

$$PRTFST(i, j, k, t) = 2\max(r_i, t + s_{ki}) + p_i + s_{ij}. \quad (3.5.3)$$

Définition 3.5.1.3. Pour toute paire de tâches $\{i, j\}$ et à tout instant t où une tâche k est complétée à la fin d'une séquence partielle π , la fonction *PRTFSS* (Priority Rule for Total Flow-time with

Symmetric Set-ups) (dans le cas où les temps logistiques sont symétriques) est définie par :

$$PRTFSS(i,k,t) = 2 \max(r_{i,t} + s_{ki}) + p_i. \quad (3.5.4)$$

Alors on a les corollaires suivantes du théorème 3.5.1.1.

Corollaire 3.5.1.4. À tout instant t où une tâche k est complétée à la fin d'une séquence partielle π , et pour toute paire de tâches $\{i, j\}$,

$$i \preceq j \text{ si et seulement si } PRTFST(i,j,k,t) \leq PRTFST(j,i,k,t). \quad (3.5.5)$$

Preuve. Ce corollaire est juste le théorème 3.5.1.1 exprimé avec la définition 3.5.1.2 de la fonction $PRTFST$. \square

Corollaire 3.5.1.5. Si les temps logistiques sont symétriques, i.e. si pour toute paire de tâches $\{i, j\}$, $s_{ij} = s_{ji}$, alors à tout instant t où une tâche k est complétée à la fin d'une séquence partielle π ,

$$i \preceq j \text{ si et seulement si } PRTFSS(i,k,t) \leq PRTFSS(j,k,t). \quad (3.5.6)$$

Preuve. L'inégalité 3.5.2 dans le théorème 3.5.1.1 est

$$i \preceq j \text{ si et seulement si } [2\max(r_{i,t} + s_{ki}) + p_i + s_{ij}] - [2\max(r_{j,t} + s_{kj}) + p_j + s_{ji}] \leq 0.$$

qui est la même chose que

$$i \preceq j \text{ si et seulement si } [2\max(r_{i,t} + s_{ki}) + p_i] - [2\max(r_{j,t} + s_{kj}) + p_j] + (s_{ij} - s_{ji}) \leq 0.$$

Et comme les temps logistiques sont symétriques i.e. $s_{ij} = s_{ji}$, alors l'inégalité précédente devient

$$i \preceq j \text{ si et seulement si } [2\max(r_{i,t} + s_{ki}) + p_i] - [2\max(r_{j,t} + s_{kj}) + p_j] \leq 0. \quad (3.5.7)$$

qui signifie, avec l'égalité 3.5.4 définissant la fonction $PRTFSS$, que

$$i \preceq j \text{ si et seulement si } PRTFSS(i,k,t) \leq PRTFSS(j,k,t). \quad \square$$

Sur la base de ces corollaires, la fonction $PRTFST$ définit une règle de priorité d'optimalité locale pour la minimisation du temps de réponse total (les tâches n'étant pas pondérées) avec des dates d'occurrence inégales des tâches, et en présence de temps logistiques dépendant de la séquence des tâches (lorsque les temps logistiques ne sont pas symétriques). Dans les mêmes conditions mais avec des temps logistiques qui sont symétriques, la fonction $PRTFSS$ définit une règle de priorité d'optimalité locale pour le temps de réponse total.

Remarque 3.5.1.6. 1) La relation de dominance \preceq dans la définition 3.4.1.2 basée sur la fonction $PRTFST$ (lorsque les temps logistiques ne sont pas symétriques), ne définit pas une relation d'ordre total dans l'ensemble des tâches. Son utilisation, tout comme celle de la fonction $PRTWF$, se fera en exploitant la matrice de dominance et le vecteur des forces de tâches.

2) En revanche, lorsque les temps logistiques sont symétriques, la relation de dominance devient une relation d'ordre total dans l'ensemble des tâches à chaque instant de décision. Dans ces conditions, la tâche fournissant la plus petite valeur de la fonction $PRTFSS$ à la complétion d'une tâche k sera prioritaire.

Remarque 3.5.1.7. Dans le cas particulier où les temps logistiques sont tous assimilés à 0 i.e. pour toute paire de tâches $\{i,j\}$ $s_{ij} = 0$, les fonctions *PRTFST* et *PRTFSS* deviennent

$$PRTFST(i,j,k,t) = 2\max(r_i,t) + p_i \text{ et } PRTFSS(i,k,t) = 2\max(r_i,t) + p_i \quad (3.5.8)$$

Ces dernières expressions dépendent seulement d'une tâche à la fois. C'est l'expression de la fonction *PRTF* (Priority Rule for Total Flow-time), règle de priorité localement optimale dans la définition 3.3.1.1 pour la minimisation du temps de réponse en présence de dates d'occurrence inégales des tâches dans le cas où il n'y a pas de temps logistiques entre les tâches.

Nous allons maintenant donner le principe de base pour l'utilisation des fonctions *PRTFST* et *PRTFSS*. Pour cela, nous considérons un ensemble de n tâches (non pondérées), avec des dates d'occurrence inégales et des temps logistiques dépendant de leur séquence, à ordonnancer sur une ressource afin de minimiser leur temps de réponse total. La procédure considérée ici est une procédure en statique.

3.5.2 Algorithmes de base

Algorithme 3.5.2.1. Principe de base du *PRTFST* : AlgoOF-*PRTFST* (temps logistiques non symétriques)

Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée à la complétion d'une tâche k , les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$
 2. Calculer $PRTFST(i,j,k,t)$ pour tous les couples de tâches (i,j) dans S .
 3. Calculer la matrice de dominance Ω dans S .
 4. Calculer le vecteur des forces (f_i) dans S .
 5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
- Tant que $\text{card}(S) > 1$, aller à l'étape 4.*

Dans cette procédure, il faut noter comme dans l'usage de la fonction *PRTWF*, qu'à chaque prise de décision, la matrice donnant les valeurs de la fonction *PRTFST* et la matrice de dominance ne sont calculées qu'une fois. Lorsqu'on réduit l'ensemble de recherche de la tâche à ordonnancer, la date de décision ne change pas et la tâche k qui est terminée est aussi la même. Ainsi les valeurs dans ces deux matrices ne changent pas. Leur restriction au nouvel ensemble de recherche est considérée. Seul le vecteur des forces des tâches change, en ne tenant compte que des tâches dans ce nouvel ensemble de recherche.

Algorithme 3.5.2.2. Principe de base du *PRTFSS* : AlgoOF-PRTFSS (temps logistiques symétriques)

Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée à la complétion d'une tâche k , les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$
2. Calculer $PRTFSS(i,k,t)$ pour toutes les tâches i dans S .
3. Déterminer la tâche fournissant la plus petite valeur de la fonction $PRTFSS$ dans S et l'ordonnancer.

Remarque 3.5.2.3. Ces algorithmes peuvent aussi être adaptés dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égal à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant le plus petit indice parmi elles. La sélection des tâches se fait suivant les mêmes étapes que précédemment.

Les complexités des algorithmes AlgoOF-PRTFST et AlgoOF-PRTFSS sont données dans la proposition suivante.

Proposition 3.5.2.4. 1) Si n est le nombre de tâches à ordonnancer, l'algorithme AlgoOF-PRTFST basé sur le principe de la fonction *PRTFST*, de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$.

2) L'algorithme AlgoOF-PRTFSS a une complexité en $O(n \cdot \log(n))$.

Preuve. 1) Dans la procédure 3.5.2.1 de l'algorithme AlgoOF-PRTFST, les étapes 1 à 5 s'exécutent en $O(n^2)$. Ainsi, chaque décision est prise en $O(n^2)$. Ces étapes sont exécutées jusqu'à la dernière tâche. Cela se fait n fois. D'où la complexité de la procédure AlgoOF-PRTFST est en $O(n^3)$.

2) Dans la procédure AlgoOF-PRTFSS, chaque décision est prise en cherchant la tâche qui donne la plus petite valeur de la fonction *PRTFSS*. Cela se fait en $O(\log(n))$ en utilisant un arbre binaire. Et puisque ces étapes sont reprises jusqu'à la dernière tâche, on a alors une complexité en $nO(\log(n))$ soit une complexité en $O(n \cdot \log(n))$ □

Dans la sous-section suivante, nous proposons une procédure qui fournit une borne inférieure du temps de réponse dans le cas de problème statique avec des temps logistiques dépendants de la séquence des tâches et des dates d'occurrence inégales des tâches.

3.5.3 Borne inférieure du temps de réponse en présence de temps logistiques dépendant de la séquence

La borne inférieure que nous proposons est appelée LB-FSDS (Lower Bound to the Flow-time with Sequence Dependent Setups). Son principe est énoncé dans la proposition 3.5.3.1 ci-après.

Proposition 3.5.3.1. LB-FSDS

Soit un ensemble de tâches $\{1, \dots, n\}$ à ordonnancer sans préemption sur une ressource. Chaque tâche i a une date d'occurrence r_i et une durée d'exécution p_i . On suppose par ailleurs qu'il y a un temps logistique s_{ij} (dépendant de la séquence des tâches) de la fin d'une tâche i au début d'une autre tâche j et qu'il y a un temps logistique initial s_{0i} avant le début de la tâche i si elle est la première tâche exécutée sur la ressource. Ces temps logistiques peuvent être symétriques ou non.

Alors, la procédure suivante (dans laquelle la préemption des temps logistiques et des tâches est permise), donne une borne inférieure du temps de réponse total des tâches ($\sum_{i=1}^n (c_i - r_i)$) où c_i est la date de complétion de la tâche i .

- Première étape : Pour chaque tâche i (i allant de 1 à n), définir $s_i = \min_{j \in \{0, \dots, n\}} (s_{ji})$. Cela devient un temps logistique vers la tâche i qui ne dépend pas de la séquence.
- Deuxième étape : À tout instant t , définir pour chaque tâche i le reste ($res(s_i)$) du temps logistique vers la tâche i et le reste ($res(p_i)$) de la durée d'exécution de la tâche i . Ordonnancer parmi les tâches éligibles, la tâche (ou le morceau de tâche) ayant la plus petite somme $res(s_i) + res(p_i)$.

Remarque 3.5.3.2. 1) Dans la procédure LB-FSDS, une seule ressource exécute toutes les tâches.

2) La procédure décrite dans LB-FSDS donne une borne inférieure aussi bien dans le cas où les temps logistiques sont symétriques que dans le cas où ils ne le sont pas.

3) La préemption étant autorisée dans cette procédure, le temps logistique vers une tâche donnée peut être exécuté séparément de la tâche elle-même.

Preuve. (Justification de la proposition 3.5.3.1)

Pour la première étape, la diminution du temps logistique vers une tâche i n'augmente pas son temps de réponse. Ainsi, le fait de remplacer les temps logistiques s_{ij} dépendant de la séquence par le temps logistique s_j qui est leur minimum pour une tâche j , n'augmente pas la valeur du temps de réponse. Cette modification va se révéler utile dans la suite.

Pour la seconde étape, étant donné que la préemption des temps logistiques et des tâches est permise, supposons qu'à un instant t , une tâche k soit complétée (ou interrompue) à la fin d'une séquence partielle π . Considérons alors deux tâches (ou morceaux de tâches) i et j pouvant être exécutées immédiatement à la suite de leur temps logistique, si ces derniers commencent à l'instant t . Alors, il peut être vérifié que la différence entre le temps de réponse de la séquence partielle πij et celui de la séquence partielle πji est

$$(2s_{ki} + p_i - s_{ji}) - (2s_{kj} + p_j - s_{ij}) \quad (3.5.9)$$

La séquence partielle πij est localement meilleure que πji pour le temps de réponse si la quantité dans l'expression 3.5.9 est négative, c'est à dire

$$(2s_{ki} + p_i - s_{ji}) \leq (2s_{kj} + p_j - s_{ij}) \quad (3.5.10)$$

Lorsque les temps logistiques sont modifiés comme dans la première étape, l'expression 3.5.10 devient

$$(2s_i + p_i - s_i) \leq (2s_j + p_j - s_j) \text{ soit } (s_i + p_i) \leq (s_j + p_j)$$

ou encore, si i et j sont plutôt des morceaux de tâches

$$res(s_i) + res(p_i) \leq res(s_j) + res(p_j) \quad (3.5.11)$$

Considérons maintenant deux ordonnancements complets π_1 et π_2 qui sont différents par la seule inversion des tâches (ou morceaux de tâches) i et j qui vérifient la condition 3.5.11 et représentés comme sur la figure 3.6.

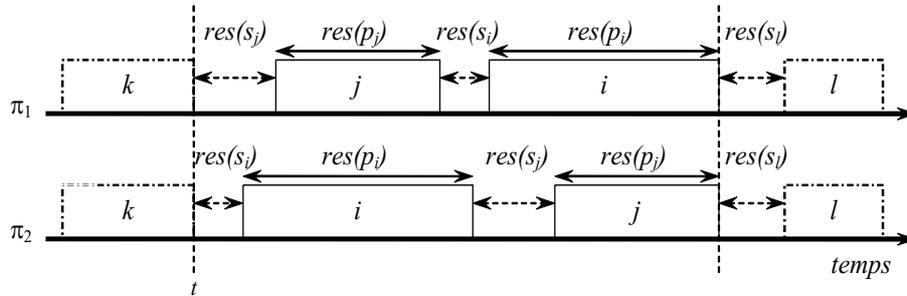


Figure 3.6 – Différence entre π_1 et π_2 avec temps logistiques

Alors la différence entre le temps de réponse de π_1 et celui de π_2 est $(res(s_i) + res(p_i)) - (res(s_j) + res(p_j))$ qui est négative. Le temps de réponse de π_1 est donc inférieur à celui de π_2 .

Ainsi, avec la préemption, l'ordonnancement dans lequel pour deux morceaux quelconques i , j de tâches contiguës (exécutées sans temps mort entre les deux), i est exécutée avant j si et seulement si $(res(s_i) + res(p_i)) \leq (res(s_j) + res(p_j))$ donne un temps de réponse optimal pour le cas préemptif. Si, dans la préemption, chaque tâche peut être exécutée sans interruption, on obtient alors une solution optimale pour le cas sans préemption (après l'étape de modification des temps logistiques). Ainsi, la solution obtenue avec la procédure LB-FSDS avec permission de la préemption donne une borne inférieure du temps de réponse dans le cas sans préemption. \square

La complexité de cette procédure LB-FSDS est donnée dans la proposition suivante.

Proposition 3.5.3.3. : Complexité de la procédure LB-FSDS

Si n est le nombre de tâches à ordonnancer, alors la procédure LB-FSDS peut être programmée avec une complexité de $O(n \cdot \log(n))$

Preuve. Les instants de décision dans la procédure LB-FSDS sont de quatre (4) types :

- lorsqu'une tâche arrive,
- lorsqu'une tâche est complétée,
- lorsqu'un temps logistique est terminé ; c'est le cas lorsque l'exécution du temps logistique est séparée de celle de la tâche en question,
- aux dates qui sont du type $\max(0, r_i - s_i)$.

Ainsi, les dates de décision sont au plus au nombre de $4n$ qui est donc en $O(n)$. Or à chaque instant de décision, la tâche donnant la plus petite somme $res(s_i) + res(p_i)$ est sélectionnée. Cela se fait, en utilisant un arbre binaire, en $O(\log(n))$. Finalement, la complexité de la procédure est en $O(n) \cdot O(\log(n))$ soit en $O(n \cdot \log(n))$. \square

Détails de la procédure LB-FSDS

Algorithme 3.5.3.4. Il faut noter que dans cette procédure, le temps logistique d'une tâche i peut être exécuté avant même sa date d'occurrence (arrivée) r_i , alors que la tâche elle-même ne peut commencer avant cette date d'arrivée.

1. Entrer les données de l'ensemble des tâches :
 - Pour chaque tâche i ($i = 1, \dots, n$) entrer la date d'arrivée r_i et la durée d'exécution p_i ;
 - Entrer les temps logistiques s_{ij} ($i = 0, \dots, n$ et $j = 1, \dots, n$).
2. Pour chaque tâche i ($i = 1, \dots, n$), déterminer le temps logistique réduit $s_i = \min_{j \in \{0, \dots, n\}} (s_{ji})$.
3. Déterminer les instants potentiels de décision :
 - Définir, pour chaque tâche $i = 1, \dots, n$, la date d'arrivée modifiée $r'_i = \max(0, r_i - s_i)$.
 - Trier les dates r_i par ordre croissant dans $R := \{R_1, \dots, R_n\}$.
 - Trier les dates r'_i par ordre croissant dans $R' := \{R'_1, \dots, R'_n\}$.
 - L'ensemble des dates de décision potentielles est $T := \{r_i, i = 1, \dots, n\} \cup \{r'_i, i = 1, \dots, n\} \cup \{\infty\}$.
 - Ordonner T par ordre croissant $T := \{T_1, \dots, T_{2n}, T_{2n+1}\}$ avec $T_{2n+1} = \infty$.
4. $t \leftarrow 0$; nombre_fini $\leftarrow 0$.
5. - Si $0 \in T$, alors choisir parmi les tâches i vérifiant $r'_i = 0$ celle qui a la plus petite somme $res(s_i) + res(p_i)$ et l'ordonnancer.
 - Sinon choisir parmi les tâches i telles que $r'_i = R'_1$, celle qui a la plus petite somme $res(s_i) + res(p_i)$ et l'ordonnancer.
6. Supposons que la tâche ordonnancée soit i .
 Tant que nombre_fini $< n$, faire
 6. 1. $t_1 \leftarrow t + res(s_i)$; $t_2 \leftarrow t + res(s_i) + res(p_i)$; $t_3 \leftarrow \min\{T_j / T_j > t\}$.
 6. 2. Si $t + res(s_i) < r_i$ (l'exécution du temps logistique est séparée de celle de la tâche i) alors
 - $t \leftarrow \min(t_1, t_3)$.
 - Si $t = t_1 < t_3$ alors terminer l'exécution de $res(s_i)$.
 Sélectionner parmi les tâches j vérifiant $r'_j \leq t$, celle qui a la plus petite somme $res(s_j) + res(p_j)$ et l'ordonnancer.
 S'il n'existe pas une telle tâche alors exécuter le temps logistique de la tâche j vérifiant $r'_j = \min\{R'_k / R'_k > t\}$.
 Aller au début de l'étape 6.
 - Si $t = t_3 \leq t_1$ alors réajuster $res(s_i)$ et ordonnancer la tâche j entrant en jeu à la place de la tâche i .
 Aller au début de l'étape 6.

6. 3. Si $t + \text{res}(s_i) \geq r_i$ (l'exécution du temps logistique n'est pas séparé de la tâche i) alors

– $t \leftarrow \min(t_2, t_3)$

– Si $t = t_2 \leq t_3$ alors terminer l'exécution de $\text{res}(s_i)$ et $\text{res}(p_i)$.

$\text{nombre_fini} \leftarrow \text{nombre_fini} + 1$.

Sélectionner parmi les tâches j vérifiant $r_j^l \leq t$, celle qui a la plus petite somme $\text{res}(s_j) + \text{res}(p_j)$ et l'ordonnancer.

S'il n'existe pas une telle tâche alors exécuter le temps logistique de la tâche j vérifiant $r_j^l = \min\{R_k^l/R_k^l > t\}$.

Aller au début de l'étape 6.

– Si $t = t_3 < t_2$ alors

Comparer la tâche i à la tâche j entrant en jeu pour décider s'il faut ou pas interrompre i au profit de j sur la base du critère $\text{res}(s_i) + \text{res}(p_i)$.

Réajuster $\text{res}(s_i)$ et $\text{res}(p_i)$.

Aller au début de l'étape 6.

Remarque 3.5.3.5. Il faut noter que :

1) De façon triviale, $\sum_{i=1}^n p_i$ est une borne inférieure du temps de réponse dans la situation présente.

2) En supposant que tous les temps logistiques sont égaux à 0, la règle du SRPT (Shortest Remaining Processing Time) fournit aussi une borne inférieure du temps de réponse dans le cas présent, et cette borne inférieure est meilleure que celle du 1).

3) La borne inférieure fournie par la procédure LB-FSDS est meilleure que celles du 1) et du 2).

Remarque 3.5.3.6. Par ailleurs :

1) La borne inférieure proposée dans la proposition 3.5.3.1 peut être judicieusement utilisée dans une procédure de branch and bound pour la résolution du problème statique de minimisation de temps de réponse en présence des temps logistiques entre les tâches et de dates d'occurrence inégales de ces tâches.

2) Compte tenu de l'étape de réduction des temps logistiques qui peut souvent réduire de façon significative les temps logistiques dans le cas de cette borne inférieure, surtout lorsqu'il y a une importante disparité entre ces temps logistiques d'une tâche à une autre, la borne inférieure fournie dans la proposition 3.5.3.1 peut être de façon plus ou moins importante, inférieure à la valeur réelle du temps de réponse dans ces cas de grande disparité entre les temps logistiques. Ainsi, il peut donc s'avérer mieux dans ces cas, d'utiliser cette borne inférieure dans ces genres de procédure, plutôt que dans un but d'évaluation de la valeur du temps de réponse fournie par un algorithme.

3.5.4 Exemples illustratifs des procédures avec des temps logistiques

Exemple 3.3 : illustration de la procédure AlgoOF-PRTFST

Les données de l'instance de l'exemple 3.3 sont en partie constituées des celles de l'exemple 3.1 dans le cas de la fonction PRTWF. Les dates d'occurrence sont reprises aussi bien que les durées

d'exécution. A ces données se rajoutent les temps logistiques non symétriques entre les tâches. Ils sont issus de générations aléatoires. Les données de cette instance sont sur les tableaux 3.9 et 3.10.

i	1	2	3	4	5
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03

Table 3.9 – Instance de l'exemple 3.3 pour les fonctions $PRTFST$ et $PRTFSS$

$\downarrow i, j \rightarrow$	1	2	3	4	5
0	0,38	0,70	0,20	2,97	0,71
1	-	0,46	0,27	1,69	0,48
2	0,28	-	0,32	0,81	0,86
3	0,71	1,53	-	0,43	0,19
4	0,53	0,29	1,71	-	1,59
5	0,28	0,04	0,09	0,26	-

Table 3.10 – Matrice des temps logistiques s_{ij} dans l'exemple 3.3

À l'instant $t = 0$, la tâche k complétée est la tâche fictive 0. Les valeurs de la fonction $PRTFST$, la matrice de dominance et le vecteur des forces des tâches à cette date sont représentées respectivement dans les tableaux 3.11, 3.12 et 3.13.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	3,50	3,31	4,73	3,52
2	8,30	-	8,34	8,83	8,88
3	7,49	8,31	-	7,21	6,97
4	6,98	6,74	8,16	-	8,04
5	3,73	3,49	3,54	3,71	-

Table 3.11 – Valeurs de la fonction $PRTFST$ à $t = 0$ dans l'exemple 3.3

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	1	1
2	0	-	0	0	0
3	0	1	-	1	0
4	0	1	0	-	0
5	0	1	1	1	-

Table 3.12 – Matrice de dominance à $t = 0$ dans l'exemple 3.3

Sur la base du vecteur des forces f_i , la tâche 1 est sélectionnée à la date $t = 0$. Les autres instants de décision et les décisions à ces instants sont regroupés dans le tableau 3.14. La séquence d'exécution des tâches est donc représentée sur le tableau 3.15 et le coût moyen (temps de réponse moyen par tâche) est 3,44. La borne inférieure à ce temps de réponse moyen fournie par la procédure LB-FSDS dans la proposition 3.5.3.1 est 2,214. La séquence d'exécution

Tâche i	1	2	3	4	5
Force f_i	4	0	2	1	3

Table 3.13 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.3

Instant de décision (t)	1,90	4,41	4,72	5,66
Tâche complétée	1	5	3	4
Ensemble des tâches	{2, 3, 4, 5}	{2, 3, 4}	{2, 4}	{2}
Tâches sélectionnée	5	3	4	2
Date de début (après set-up)	2,38	4,50	5,15	5,95

Table 3.14 – Autres décisions dans l'exemple 3.3 pour la fonction PRTFST

des morceaux des tâches et de leurs temps logistiques est dans le tableau 3.16. Dans ce dernier tableau, les morceaux des tâches réelles et leurs dates de début et fin sont représentés en gras. Les morceaux des temps logistiques quant à eux ne sont pas notés en gras.

Ordre d'exécution	1	2	3	4	5
Tâche	1	5	3	4	2
Date de début	1,14	2,38	4,50	5,15	5,95
Date de fin	1,90	4,41	4,72	5,66	8,89

Table 3.15 – Séquence d'exécution des tâches de l'exemple 3.3

Ordre d'exécution	1	2	3	4	5	6	7
Tâche	5	1	5	4	4	1	1
Date de début	0	0,19	0,28	0,92	1,18	1,69	1,88
Date de fin/interruption	0,19	0,28	0,92	1,18	1,69	1,88	2,64
Ordre d'exécution	8	9	10	11	12	13	
Tâche	5	3	3	5	2	2	
Date de début	2,64	3,19	3,28	3,50	4,34	4,38	
Date de fin/interruption	3,19	3,28	3,50	4,34	4,38	7,32	

Table 3.16 – Séquence d'exécution des tâches dans la borne inférieure de l'exemple 3.3

Exemple 3.4 : illustration de la procédure AlgoOF-PRTFSS

Dans l'instance de l'exemple 3.4, en dehors de la matrice des temps logistiques, les données sont les mêmes que dans l'exemple 3.3 dans le tableau 3.9. La matrice des temps logistiques est constituée de la partie triangulaire supérieure de celle de l'exemple 3.3 dans le tableau 3.10.

À l'instant $t = 0$, la tâche k complétée est la tâche fictive 0. Les valeurs de la fonction PRTFSS pour à cette date sont représentées dans le tableau 3.18.

Sur la base de ces valeurs de la fonction PRTFSS, la tâche 1 est sélectionnée à la date $t = 0$. Les autres décisions dans cet exemple sont représentées sur le tableau 3.19.

La valeur du temps de réponse moyen fourni par cette séquence de tâches est de 3,342. La valeur de la borne inférieure fournie par la procédure LB-FSDS peut être retrouvée de la même

$\downarrow i, j \rightarrow$	1	2	3	4	5
0	0,38	0,70	0,20	2,97	0,71
1	-	0,46	0,27	1,69	0,48
2	0,46	-	0,32	0,81	0,86
3	0,27	0,32	-	0,43	0,19
4	1,69	0,81	0,43	-	1,59
5	0,48	0,86	0,19	1,59	-

Table 3.17 – Matrice des temps logistiques s_{ij} dans l'exemple 3.4

Tâche i	1	2	3	4	5
$PRTFSS(i,k,t)$	3,04	8,02	6,78	6,45	3,45

Table 3.18 – Valeurs de la fonction $PRTFSS$ des tâches à $t = 0$ dans l'exemple 3.4

Instant de décision (t)	1,90	4,41	4,72	5,66
Tâche complétée	1	5	3	4
Ensemble des tâches	{2, 3, 4, 5}	{2, 3, 4}	{2, 4}	{2}
Tâches sélectionnée	5	3	4	2
Date de début (après set-up)	2,38	4,50	5,00	5,80

Table 3.19 – Autres décisions dans l'exemple 3.4 pour la fonction $PRTFSS$

façon que dans l'exemple 3.3.

Sur la base de tout ce qui précède, nous reconsidérons maintenant problème (\mathcal{P}_{13}) dans sa forme temps-réel.

3.5.5 Algorithmes en temps-réel pour la résolution du problème (\mathcal{P}_{13})

En plus de tous les développements ci-dessus, la proposition suivante nous permettra par la suite, comme dans les problèmes (\mathcal{P}_{11}) et (\mathcal{P}_{12}), de parler indifféremment des entités que des demandes de tâches de maintenance sur ces entités. La justification de cette proposition est analogue à celle de la proposition 3.3.1.3.

Proposition 3.5.5.1. *En respectant les corollaires 3.5.1.4 et 3.5.1.5 avec les fonctions $PRTFST$ et $PRTFSS$, on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} . En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, les corollaires 3.5.1.4 et 3.5.1.5 permettent de programmer la $m_1^{i\text{eme}}$ tâche sur \mathcal{E}_{ik} avant la $m_2^{i\text{eme}}$.*

Ainsi, dans la suite de cette section, nous parlerons indifféremment de sélection d'une tâche sur une entité que de sélection d'une entité. Dans la procédure en temps-réel, nous utiliserons en plus des fonctions $PRTFST$ et $PRTFSS$, des corollaires 3.5.1.4 et 3.5.1.5 et de la proposition 3.5.5.1, le concept de tâches urgentes introduite dans la définition 3.3.1.4.

Partant de tout ce qui précède, les procédures suivantes (deux procédures sont représentées dans cet algorithme 3.5.5.2) sont proposées pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{13}).

Algorithme 3.5.5.2. Procédures en temps-réel OL-MTSA-ST et OL-MTSA-SS

La procédure prend le nom OL-MTSA-ST lorsque les temps logistiques ne sont pas symétriques et OL-MTSA-SS lorsqu'ils le sont.

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques sur chaque entité du système comme définies dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer, suivant que les temps logistiques sont symétriques ou non, la procédure décrite dans l'algorithme 3.5.2.1 (AlgoOF-PRTFST) ou dans l'algorithme 3.5.2.2 (AlgoOF-PRTFSS) à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que c'est l'entité \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

La proposition suivante donne la complexité de la procédure ci-dessus.

Proposition 3.5.5.3. Complexité

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-ST est en $O\left(N' \left(\sum_{k=1}^N N_k\right)^2\right)$. Dans ces mêmes conditions, la complexité de la procédure OL-MTSA-SS est en $O\left(N' \left(\sum_{k=1}^N N_k\right)\right)$.

La justification de cette proposition pour la procédure OL-MTSA-ST est analogue à celle de la complexité de la procédure OL-MTSA-W dans la proposition 3.4.5.3. Celle de la procédure OL-MTSA-SS est analogue à la complexité de la procédure OL-MTSA dans la proposition 3.3.2.2.

3.5.6 Résultats numériques**Expérimentations**

Pour les expérimentations, nous reconsidérons le protocole d'expérimentation décrit dans la sous-section 3.3.3 en considérant à présent les données relatives aux temps logistiques entre les sites,

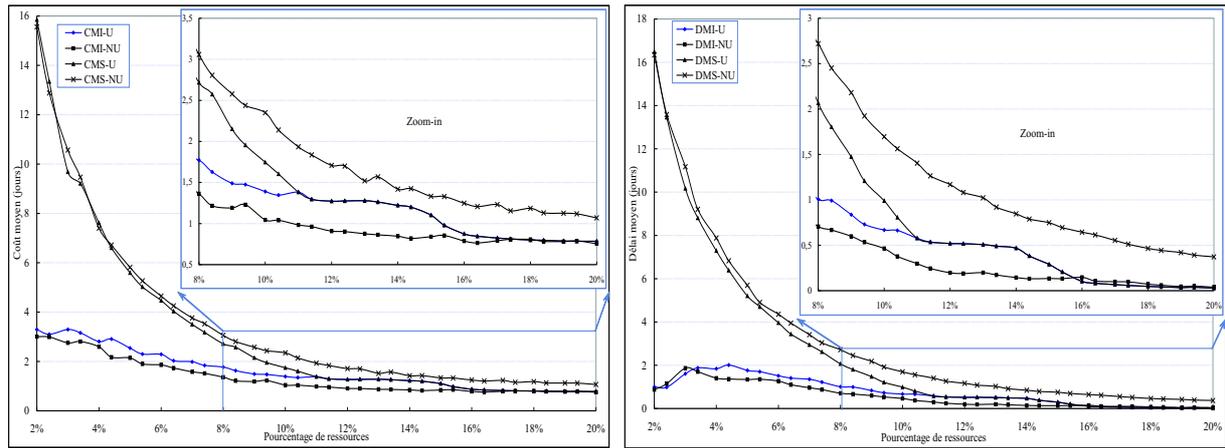
mais pas celles relatives aux poids des entités, ces poids étant tous égaux à 1 ici. Les résultats d'expérimentation montrés ici sont relatifs, sur le jeu de données décrites dans la sous-section 3.3.3, à un ensemble de 10 sites. Le nombre moyen d'entités par site étant de 50, le nombre moyen d'entités dans le système composé des 10 sites est de 500 entités. Mais le nombre réel d'entités par site étant aléatoire uniforme entre 20 et 80, le nombre réel d'entités dénombrées dans le système des 10 sites est de 441 entités. Le lecteur pourra se référer à la sous-section 3.3.3 pour la description des données. Les données utilisées concernant des temps logistiques symétriques entre les sites, la procédure avec la fonction PRTFSS est utilisée ici plutôt que celle avec la fonction PRTFST.

Par ailleurs, comme dans les sections précédentes, nous avons fait varier ici aussi le nombre total de ressources de maintenance pour tout le système en proportion du nombre total (son estimation étant de 500) d'entités dans le système. Ce nombre de ressources varie entre 2% et 20% du nombre total estimé d'entités variant donc entre 10 et 100 ressources. L'exécution des programmes se fait sur un horizon de 365 jours. Les résultats sont représentés en fonction de la proportion du nombre total d'entités que représente le nombre de ressources de maintenance. La synthèse des résultats est présentée sur la figure 3.7. Nous gardons sur cette figure les mêmes appellations des paramètres que dans les sections précédentes. Les coûts des états critiques CMI et CMS représentent les temps de réponse moyens de tâches de maintenance. Par ailleurs, trois types de courbes sont représentés sur les temps moyens par ressources. En plus de TMTpR qui représente le temps moyen de travail des ressources (en activité sur les entités), TMDpR présente le temps moyen de déplacement par ressource et TMGpR est le temps moyen global par ressource (en activité au niveau des entités et en déplacement). Chacun de ces paramètres est relevé lorsque le concept d'urgence est utilisé dans la prise de décision (U), et (NU) lorsqu'il ne l'est pas.

Analyse des résultats

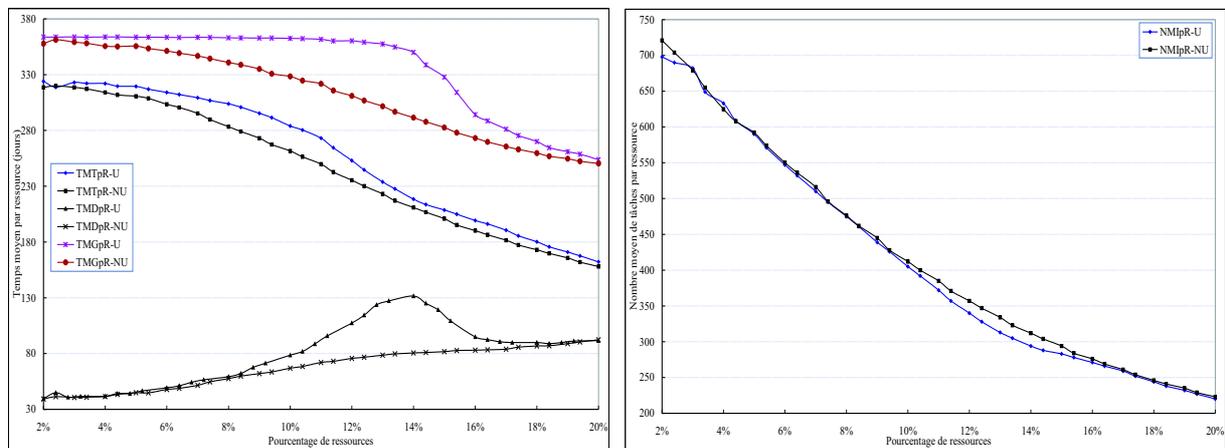
Pour les courbes 3.7(a), 3.7(b) et 3.7(d), les observations sont analogues à celles de la sous-section 3.3.4 sauf qu'ici, la convergence des coûts CMI et CMS (et les délais DMI et DMS) est réalisée à 11% au lieu de 10% lorsqu'il n'y avait pas de temps logistiques considérables. Sur la courbe 3.7(c), on a les mêmes observations les temps global moyen par ressource TMGpR que celles qu'on avait sur le temps d'activité par ressource TMTpR lorsqu'il n'y avait pas de temps logistiques considérables. En outre, on voit sur cette sous-figure que le temps moyen de déplacement par ressource lorsque le concept d'urgence est utilisé, est légèrement supérieur au cas où ce concept n'est pas utilisé (cette différence est relativement importante entre 10% et 16% de ressources). Pour davantage d'analyse sur les résultats dans cette configuration, le lecteur peut se référer par exemple à [2].

Considérons maintenant, en plus des temps logistiques entre les sites, que les entités des différents sites ont des poids inégaux. C'est la configuration (\mathcal{P}_{14}) du problème (\mathcal{P}_1) qui est l'objet de la section qui suit.



(a) Coût moyen de maintenance (en temps de réponse)

(b) Délai moyen d'intervention



(c) Temps moyen d'activité par ressource

(d) Nombre moyen de tâches par ressource

Figure 3.7 – Résultats avec des poids égaux et des temps logistiques importants

3.6 Configuration avec des entités pondérées et des temps logistiques considérables

Cette section est consacrée à la configuration (\mathcal{P}_{14}) du problème (\mathcal{P}_1). Dans cette configuration, non seulement les temps logistiques entre les sites sont considérables comme dans la section précédente et ne permettraient pas d'hypothèse réductrice, mais aussi les différentes entités de chaque site ont des poids inégaux. Les généralités sur cette configuration, en dehors des poids des entités du système, sont les mêmes que dans la section précédente. Si w_{ik} est le poids de l'entité \mathcal{E}_{ik} , le coût des états critiques à minimiser sur l'horizon de planification \mathcal{H} , en présence des temps logistiques entre les sites, est

$$C_{ot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} w_{ik}(c_{m,ik} - r_{m,ik}) \right) \quad (3.6.1)$$

Comme dans les cas précédents, nous commençons par une analyse sur un problème localement statique afin de trouver des outils de décision.

3.6.1 Le problème statique et les règles de décision

Nous n'allons pas reprendre ici les détails du problème localement statique. Nous avons un ensemble de n tâches à ordonnancer à une date t sur une ressource. Les caractéristiques de ces tâches sont les mêmes que dans la sous-section 3.5.1. Par ailleurs, chaque tâche i a un poids w_i (poids qui ne sont pas égaux pour toutes les tâches). L'objectif est de minimiser le temps de réponse pondéré total de ces tâches qui est $\sum_{i=1}^n w_i(c_i - r_i)$ si la date de complétion de la tâche i est c_i . Afin de trouver des règles de décision, nous posons la définition suivante.

Définition 3.6.1.1. *Pour toute paire de tâches $\{i, j\}$ et à tout instant t où une tâche k est complétée, la fonction PRTWFST (Priority Rule for Total Weighted Flow-time with Set-up Times) est définie comme suit*

$$PRTWFST(i, j, k, t) = (w_i + w_j) \max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}). \quad (3.6.2)$$

On a alors le théorème ci-après.

Théorème 3.6.1.2. *À tout instant t si k est la dernière tâche exécutée par la ressource dans une séquence partielle π (complétée à la date t), alors pour toute paire de tâches $\{i, j\}$, $i \preceq j$ si et seulement si*

$$PRTWFST(i, j, k, t) \leq PRTWFST(j, i, k, t) \quad (3.6.3)$$

Preuve. Soient i et j deux tâches à exécuter à la fin d'une séquence partielle π terminée par la tâche k . La tâche i a les paramètres r_i , p_i , w_i et un temps logistique s_{ki} à partir de la fin de la tâche k au début de la tâche i ; la tâche j a les paramètres r_j , p_j , w_j et un temps logistique s_{kj} à partir de la fin de la tâche k au début de la tâche j . Le temps logistique de la tâche i à la tâche j est s_{ij} et celui de la tâche j à la tâche i est s_{ji} . Ces temps temps logistiques peuvent être symétriques ou non.

Supposons que $r_i \leq r_j$. Alors il peut être vérifié en ordonnant les tâches i et j à l'instant t à la fin de la tâche k , que les assertions suivantes sont satisfaites.

i) Si $r_i \leq r_j \leq t$, alors

$$i \preceq j \text{ si et seulement si } [(w_i + w_j)s_{ki} + w_j(p_i + s_{ij})] - [(w_i + w_j)s_{kj} + w_i(p_j + s_{ji})] \leq 0.$$

ii-a) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

ii-b1) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$$i \preceq j \text{ si et seulement si } [(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)r_j + w_i(p_j + s_{ji})] \leq 0.$$

ii-b2) Si $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$$i \preceq j \text{ si et seulement si } [(w_i + w_j)s_{ki} + w_j(p_i + s_{ij})] - [(w_i + w_j)s_{kj} + w_i(p_j + s_{ji})] \leq 0.$$

iii-a1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

iii-a2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$, alors $i \preceq j$ est évident.

iii-b1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$$i \preceq j \text{ si et seulement si } [(w_i + w_j)r_i + w_j(p_i + s_{ij})] - [(w_i + w_j)r_j + w_i(p_j + s_{ji})] \leq 0.$$

iii-b2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$i \preceq j$ si et seulement si $[(w_i + w_j)r_i + w_j(p_i + s_{ij})] - [(w_i + w_j)(t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0$.

iii-c1) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} \leq r_j$, alors

$i \preceq j$ si et seulement si $[(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)r_j + w_i(p_j + s_{ji})] \leq 0$.

iii-c2) Si $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} > r_j$ et $t + s_{kj} > r_j$, alors

$i \preceq j$ si et seulement si $[(w_i + w_j)s_{ki} + w_j(p_i + s_{ij})] - [(w_i + w_j)s_{kj} + w_i(p_j + s_{ji})] \leq 0$.

Il ressort des différentes conditions et inégalités que dans les cas i), ii-b1), ii-b2), iii-b1), iii-b2), iii-c1) et iii-c2), l'on a $i \preceq j$ si et seulement si

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

Cas ii-a) : $r_i \leq t \leq r_j$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$,

Dans ce cas, la seconde inégalité est équivalente à

$$(w_i + w_j)(t + s_{ki}) + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j.$$

Cela implique que

$$(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)(t + s_{ki}) + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Les premier et deuxième membres de cette dernière chaîne d'inégalités donnent

$$(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Cette inégalité est équivalente à celle qui suit, si nous considérons la première condition dans le cas ii-a)

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji})$$

de laquelle nous déduisons que

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}) \leq (w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji}).$$

Enfin, le premier et le dernier membres de cette chaîne d'inégalités impliquent

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

Cas iii-a1) : $t \leq r_i \leq r_j$ et $t + s_{ki} \leq r_i$ et $r_i + p_i + s_{ij} \leq r_j$.

En multipliant la troisième inégalité par la quantité positive $(w_i + w_j)$, on obtient

$$(w_i + w_j)r_i + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j,$$

ce qui implique

$$(w_i + w_j)r_i + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_i + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Le premier et le dernier membres de cette chaîne d'inégalités entraînent

$$(w_i + w_j)r_i + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Ainsi, en considérant cette inégalité et la deuxième condition dans le cas iii-a1), l'on obtient

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Cette inégalité donne alors

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}) \leq (w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji}).$$

Les membres extrêmes de cette chaîne d'inégalités donnent finalement l'inégalité suivante

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

Cas iii-a2) : $t \leq r_i \leq r_j$ et $t + s_{ki} > r_i$ et $t + s_{ki} + p_i + s_{ij} \leq r_j$.

La troisième inégalité donne, en la multipliant par la quantité positive $(w_i + w_j)$,

$$(w_i + w_j)(t + s_{ki}) + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j.$$

Ceci donne

$$(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)(t + s_{ki}) + (w_i + w_j)(p_i + s_{ij}) \leq (w_i + w_j)r_j \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Et nous avons donc, avec les membres extrêmes de cette chaîne d'inégalités,

$$(w_i + w_j)(t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}).$$

Cette dernière inégalité implique, conjointement avec la seconde inégalité dans le cas iii-a2),

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}),$$

qui donne

$$(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij}) \leq (w_i + w_j)r_j + w_i(p_j + s_{ji}) \leq (w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji}).$$

D'où nous obtenons finalement (avec le premier et le dernier membres) l'inégalité suivante

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

Ainsi, dans tous les cas où $r_i \leq r_j$, nous avons $i \preceq j$ si et seulement si

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

Supposons maintenant que $r_i > r_j$. En inversant les rôles de i et j dans l'expression précédente, nous avons : $j \preceq i$ si et seulement si

$$[(w_i + w_j)\max(r_j, t + s_{kj}) + w_j(p_j + s_{ji})] - [(w_i + w_j)\max(r_i, t + s_{ki}) + w_i(p_i + s_{ij})] \leq 0.$$

Ceci revient à : $i \preceq j$ si et seulement si

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

En conclusion, dans tous les cas, on a $i \preceq j$ si et seulement si

$$[(w_i + w_j)\max(r_i, t + s_{ki}) + w_j(p_i + s_{ij})] - [(w_i + w_j)\max(r_j, t + s_{kj}) + w_i(p_j + s_{ji})] \leq 0.$$

qui s'écrit, avec l'égalité 3.6.2 dans la définition de la fonction $PRTWFST$, comme

$$PRTWFST(i, j, k, t) \leq PRTWFST(j, i, k, t). \quad \square$$

Partant de ce théorème 3.6.1.2, la fonction $PRTWFST$ définit une règle de priorité d'optimalité locale pour la minimisation du temps total de réponse pondéré (les tâches ayant des poids inégaux), avec des dates d'occurrence inégales des tâches et en présence de temps logistiques dépendant de la séquence des tâches (les temps logistiques peuvent être symétriques ou non).

Remarque 3.6.1.3. *La relation de dominance \preceq dans la définition 3.4.1.2 basée sur la fonction $PRTWFST$ ne définit pas une relation d'ordre total dans l'ensemble des tâches à une date de décision. Son utilisation, tout comme celle des fonctions $PRTWF$ et $PRTFST$, se fera en exploitant la matrice de dominance et le vecteur des forces de tâches.*

Remarque 3.6.1.4. *Dans la fonction $PRTWFST$, il faut remarquer ce qui suit :*

1) *Dans le cas particulier où les temps logistiques sont tous assimilés à 0 i.e. pour toute paire de tâches $\{i, j\}$ $s_{ij} = 0$, et les poids des tâches sont tous égaux à 1, l'expression de la fonction $PRTWFST$ devient*

$$PRTWFST(i, j, k, t) = 2\max(r_i, t) + p_i \quad (3.6.4)$$

qui ne dépend pas de la tâche k qui est terminée et est fonction d'une seule tâche à la fois. C'est l'expression de la fonction *PRTF* utilisée dans la section 3.3.

2) Dans le cas particulier où les logistiques sont tous assimilés à 0 i.e. pour toute paire de tâches $\{i,j\}$ $s_{ij} = 0$, et les tâches ont des poids inégaux, l'expression de la fonction *PRTWFST* devient

$$PRTWFST(i,j,k,t) = (w_i + w_j)\max(r_i,t) + w_j p_i \quad (3.6.5)$$

qui ne dépend pas de la tâche k qui est terminée. C'est l'expression de la fonction *PRTWF* utilisée dans la section 3.4.

3) Dans le cas particulier où les temps logistiques sont non nuls, et les tâches ont des poids tous égaux à 1, l'expression de la fonction *PRTWFST* devient

$$PRTWFST(i,j,k,t) = 2\max(r_i,t + s_{ki}) + p_i + s_{ij} \quad (3.6.6)$$

C'est l'expression de la fonction *PRTFST* utilisée dans la section 3.5 et qui conduit à la fonction *PRTFSS* lorsque les temps logistiques sont en plus symétriques.

La figure 3.8 présente les relations entre les différentes règles d'optimalité locale utilisées.

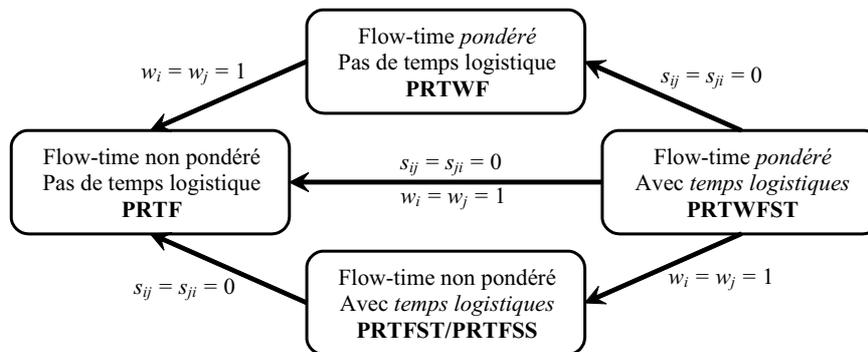


Figure 3.8 – Les différentes règles pour la minimisation du temps de réponse

Nous allons maintenant donner le principe de base pour l'utilisation de la fonction *PRTWFST*.

3.6.2 Algorithme de base pour l'utilisation de la fonction *PRTWFST*

Nous considérons un ensemble de n tâches pondérées (avec des poids inégaux) avec des dates d'occurrence inégales, et des temps logistiques dépendant de la séquence des tâches, à ordonnancer sur une ressource afin de minimiser leur temps de réponse pondéré total. La procédure considérée ici est une procédure en statique.

Algorithme 3.6.2.1. Principe de base du *PRTWFST* : AlgoOF-*PRTWFST*

Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée à la complétion d'une tâche k , les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$
2. Calculer $PRTWFST(i,j,k,t)$ pour tous les couples de tâches (i,j) dans S .
3. Calculer la matrice de dominance Ω dans S .
4. Calculer le vecteur des forces (f_i) dans S .
5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
 Tant que $\text{card}(S) > 1$, aller à l'étape 4.

Pour les mêmes raisons que dans l'usage de la fonction PRTFST, la matrice donnant les valeurs de la fonction PRTWFST et la matrice de dominance ne sont calculées qu'une seule fois à chaque prise de décision. Seul le vecteur des forces des tâches change suivant la restriction de l'ensemble de recherche de la tâche.

Remarque 3.6.2.2. Cet algorithme 3.6.2.1 peut aussi être adapté dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égale à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant le plus petit indice parmi elles. La recherche des tâches se fait suivant les mêmes étapes que précédemment.

La complexité de l'algorithme AlgoOF-PRTWFST est donnée dans la proposition suivante.

Proposition 3.6.2.3. Si n est le nombre de tâches à ordonnancer, l'algorithme AlgoOF-PRTWFST basé sur le principe de la fonction PRTWFST, de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$.

Preuve. Dans la procédure AlgoOF-PRTWFST 3.6.2.1, les étapes 1 à 5 s'exécutent en $O(n^2)$. Ainsi, chaque décision est prise en $O(n^2)$. Ces étapes sont exécutées jusqu'à la dernière tâche. Cela se fait n fois. D'où la complexité de la procédure AlgoOF-PRTWFST est en $O(n^3)$. \square

Une borne inférieure du temps de réponse pondéré en présence de date d'occurrence inégales des tâches et de temps logistiques dépendant de la séquence de ces tâches est fournie ci-après.

3.6.3 Borne inférieure du temps de réponse pondéré en présence de temps logistiques dépendant de la séquence

La procédure donnant cette borne inférieure est appelée LB-WFSDS (Lower Bound to the Weighted Flow-time with Sequence Dependent Setups). Son principe est résumé dans la proposition suivante.

Proposition 3.6.3.1. : LB-WFSDS

Soit un ensemble de tâches $\{1, \dots, n\}$ à ordonnancer sans préemption sur une ressource. Chaque tâche i a une date d'occurrence r_i , une durée d'exécution p_i et un poids w_i . On suppose qu'il y a un temps logistique s_{ij} (dépendant de la séquence) de la fin d'une tâche i au début d'une autre

tâche j et qu'il a un temps logistique initial s_{0i} avant le début de la tâche i si elle est la première tâche exécutée sur la ressource. Ces temps logistiques peuvent être symétriques ou non.

Alors la procédure suivante (dans laquelle la préemption des temps logistiques et des tâches est permise), donne une borne inférieure du temps de réponse pondéré total des tâches ($\sum_{i=1}^n w_i(c_i - r_i)$) où c_i est la date de complétion de la tâche i .

- Première étape : Pour chaque tâche i (i allant de 1 à n), définir $s_i = \min_{j \in \{0, \dots, n\}} (s_{ji})$. Cela devient un temps logistique vers la tâche i qui ne dépend pas de la séquence.
- Deuxième étape : À tout instant t , définir pour chaque tâche i le reste ($res(s_i)$) du temps logistique vers la tâche i et le reste $res(p_i)$ de la durée d'exécution de la tâche i . Ordonnancer parmi les tâches éligibles, la tâche (ou le morceau de tâche) ayant le plus petit rapport $\frac{res(s_i) + res(p_i)}{w_i}$.

Remarque 3.6.3.2. 1) Dans la procédure LB-WFSDS, une seule ressource exécute toutes les tâches.

2) La procédure décrite dans LB-WFSDS donne une bonne inférieure aussi bien dans le cas où les temps logistiques sont symétriques que dans le cas où ils ne sont pas symétriques.

3) La préemption étant autorisée dans cette procédure, le temps logistique vers une tâche donnée peut être exécuté séparément de la tâche elle-même.

Preuve. (Justification de la proposition 3.6.3.1)

Pour la première étape (tout comme dans le cas où les tâches ne sont pas pondérées), la diminution du temps logistique vers une tâche i n'augmente pas son temps de réponse. Ainsi, le fait de remplacer les temps logistiques s_{ij} dépendant de la séquence par le temps logistique s_j qui est leur minimum pour une tâche j , n'augmente pas la valeur du temps de réponse pondéré. Cette modification va se révéler utile dans la suite pour la borne inférieure.

Pour la seconde étape, étant donné que la préemption des temps logistiques et des tâches est permise, supposons qu'à un instant t , une tâche k soit complétée (ou interrompue) à la fin d'une séquence partielle π . Considérons alors deux tâches (ou morceaux de tâches) i et j pouvant être exécutées immédiatement à la suite de leur temps logistique, si ces derniers commencent à l'instant t . Alors, il peut être vérifié que la différence entre le temps de réponse pondéré de la séquence partielle πij et celui de la séquence partielle πji est

$$((w_i + w_j)s_{ki} + w_j p_i - w_i s_{ji}) - ((w_i + w_j)s_{kj} + w_i p_j - w_j s_{ij}) \quad (3.6.7)$$

La séquence partielle πij est localement meilleure que πji pour le temps de réponse pondéré si la quantité dans l'expression 3.6.7 est négative, c'est à dire

$$((w_i + w_j)s_{ki} + w_j p_i - w_i s_{ji}) \leq ((w_i + w_j)s_{kj} + w_i p_j - w_j s_{ij}) \quad (3.6.8)$$

Lorsque les temps logistiques sont modifiés comme dans la première étape, l'expression 3.6.8 devient

$$((w_i + w_j)s_i + w_j p_i - w_i s_i) \leq ((w_i + w_j)s_j + w_i p_j - w_j s_j) \text{ soit } w_j(s_i + p_i) \leq w_i(s_j + p_j)$$

ou encore, si i et j sont plutôt des morceaux de tâches

$$w_j(res(s_i) + res(p_i)) \leq w_i(res(s_j) + res(p_j)) \text{ soit } \frac{res(s_i) + res(p_i)}{w_i} \leq \frac{res(s_j) + res(p_j)}{w_j} \quad (3.6.9)$$

Considérons maintenant deux ordonnancements complets π_3 et π_4 qui sont différents par la seule inversion des tâches (ou morceaux de tâches) i et j qui vérifient la condition 3.6.9 et représentés comme sur la figure 3.9.

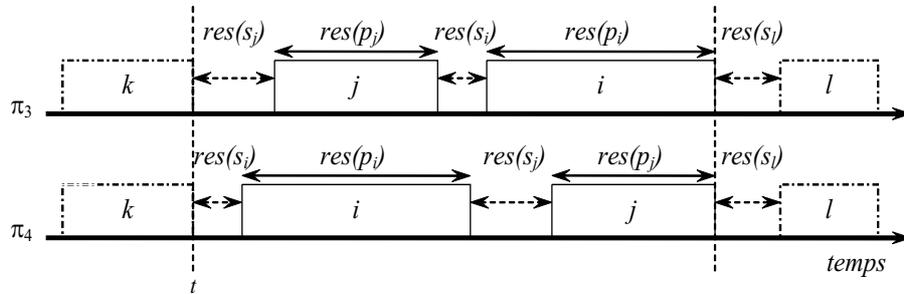


Figure 3.9 – Différence entre π_3 et π_4 avec temps logistiques et poids

Alors la différence entre le temps de réponse pondéré de π_3 et celui de π_4 est $w_j(res(s_i) + res(p_i)) - w_i(res(s_j) + res(p_j))$ qui est négative. Le temps de réponse pondéré de π_3 est donc inférieur à celui de π_4 .

Ainsi, avec la préemption, l'ordonnancement dans lequel pour deux morceaux quelconques i, j de tâches contiguës (exécutées sans temps mort entre les deux), i est exécutée avant j si et seulement si $\frac{res(s_i) + res(p_i)}{w_i} \leq \frac{res(s_j) + res(p_j)}{w_j}$, donne un temps de réponse pondéré optimal pour le cas préemptif. Si dans la préemption chaque tâche peut être exécutée sans interruption, on obtient alors une solution optimale pour le cas sans préemption après l'étape de modification des temps logistiques. Ainsi, la solution obtenue avec la procédure LB-WFSDS avec permission de la préemption donne une bonne inférieure du temps de réponse dans le cas sans préemption. \square

Proposition 3.6.3.3. : Complexité de la procédure LB-WFSDS

Si n est le nombre de tâches à ordonnancer, alors la procédure LB-WFSDS peut être programmée avec une complexité de $O(n \cdot \log(n))$

Preuve. Les instants de décisions dans la procédure LB-WFSDS sont (comme dans le cas où les tâches ne sont pas pondérées), de quatre (4) types :

- lorsqu'une tâche arrive,
- lorsqu'une tâche est complétée,
- lorsqu'un temps logistique est terminé ; c'est le cas lorsque l'exécution du temps logistique est séparée de celle de la tâche en question,
- aux dates qui sont du type $\max(0, r_i - s_i)$.

Ainsi, les dates de décision sont au plus au nombre de $4n$ qui est donc en $O(n)$. Or à chaque instant de décision, la tâche donnant le plus petit rapport $\frac{res(s_i) + res(p_i)}{w_i}$ est sélectionnée. Cela se

fait, en utilisant un arbre binaire, en $O(\log(n))$. Finalement, la complexité de la procédure est en $O(n) \cdot O(\log(n))$ soit en $O(n \cdot \log(n))$. \square

Détails de la procédure LB-WFSDS

Algorithme 3.6.3.4. Il faut noter que dans cette procédure, le temps logistique d'une tâche i peut être exécuté avant même sa date d'occurrence r_i alors que la tâche elle-même ne peut commencer avant cette date d'arrivée.

1. Entrer les données de l'ensemble des tâches :
 - Pour chaque tâche i ($i = 1, \dots, n$) entrer la date d'arrivée r_i , la durée d'exécution p_i et le poids w_i .
 - Entrer les temps logistiques s_{ij} ($i = 0, \dots, n$ et $j = 1, \dots, n$).
2. Pour chaque tâche i ($i = 1, \dots, n$), déterminer le temps logistique réduit $s_i = \min_{j \in \{0, \dots, n\}} (s_{ji})$.
3. Déterminer les instants potentiels de décision :
 - Définir, pour chaque tâche $i = 1, \dots, n$, la date d'occurrence modifiée $r_i^l = \max(0, r_i - s_i)$.
 - Trier les dates r_i par ordre croissant dans $R := \{R_1, \dots, R_n\}$.
 - Trier les dates r_i^l par ordre croissant dans $R^l := \{R_1^l, \dots, R_n^l\}$.
 - L'ensemble des dates de décision potentielles est $T := \{r_i, i = 1, \dots, n\} \cup \{r_i^l, i = 1, \dots, n\} \cup \{\infty\}$.
 - Ordonner T par ordre croissant $T := \{T_1, \dots, T_{2n}, T_{2n+1}\}$ avec $T_{2n+1} = \infty$.
4. $t \leftarrow 0$; nombre_fini $\leftarrow 0$.
5. - Si $0 \in T$, alors choisir parmi les tâches i vérifiant $r_i^l = 0$ celle qui a le plus petit rapport $\frac{res(s_i) + res(p_i)}{w_i}$ et l'ordonner.
 - Sinon choisir parmi les tâches i telles que $r_i^l = R_1^l$, celle qui a le plus petit rapport $\frac{res(s_i) + res(p_i)}{w_i}$ et l'ordonner.
6. Supposons que la tâche ordonnancée soit i .
 Tant que nombre_fini $< n$, faire
 6. 1. $t_1 \leftarrow t + res(s_i)$; $t_2 \leftarrow t + res(s_i) + res(p_i)$; $t_3 \leftarrow \min\{T_j / T_j > t\}$.
 6. 2. Si $t + res(s_i) < r_i$ (l'exécution du temps logistique est séparée de celle de la tâche i) alors
 - $t \leftarrow \min(t_1, t_3)$.
 - Si $t = t_1 < t_3$ alors terminer l'exécution de $res(s_i)$.
 Sélectionner parmi les tâches j vérifiant $r_j^l \leq t$, celle qui a le plus petit rapport $\frac{res(s_j) + res(p_j)}{w_j}$ et l'ordonner.
 S'il n'existe pas une telle tâche alors exécuter le temps logistique de la tâche j vérifiant

<p> $r'_j = \min\{R'_k/R'_k > t\}$. <i>Aller au début de l'étape 6.</i> </p> <ul style="list-style-type: none"> – Si $t = t_3 \leq t_1$ alors réajuster $res(s_i)$ et ordonnancer la tâche j entrant en jeu à la place de la tâche i. <i>Aller au début de l'étape 6.</i> <p> 6. 1. Si $t + res(s_i) \geq r_i$ (l'exécution du temps logistique n'est pas séparée de la tâche i) alors </p> <ul style="list-style-type: none"> – $t \leftarrow \min(t_2, t_3)$ – Si $t = t_2 \leq t_3$ alors terminer l'exécution de $res(s_i)$ et $res(p_i)$. $nombre_fini \leftarrow nombre_fini + 1$. <i>Sélectionner parmi les tâches j vérifiant $r'_j \leq t$, celle qui a le plus petit rapport $\frac{res(s_j) + res(p_j)}{w_i}$ et l'ordonnancer;</i> <i>S'il n'existe pas une telle tâche alors exécuter le temps logistique de la tâche j vérifiant $r'_j = \min\{R'_k/R'_k > t\}$.</i> <i>Aller au début de l'étape 6.</i> – Si $t = t_3 < t_2$ alors <i>Comparer la tâche i à la tâche j entrant en jeu pour décider s'il faut ou pas interrompre i au profit de j sur la base du critère $\frac{res(s_i) + res(p_i)}{w_i}$.</i> <i>Réajuster $res(s_i)$ et $res(p_i)$.</i> <i>Aller au début de l'étape 6.</i>
--

Remarque 3.6.3.5. *Il faut noter que :*

1) *De façon triviale, $\sum_{i=1}^n w_i p_i$ est une borne inférieure du temps de réponse dans la situation présente.*

2) *En supposant que tous les temps logistiques sont égaux à 0, la règle du SWRPT (Shortest Weighted Remaining Processing Time) fournit aussi une borne inférieure du temps de réponse dans le cas présent, et cette borne inférieure est meilleure que celle du 1).*

3) *La borne inférieure fournit par la procédure LB-WFSDS est meilleure que celles du 1) et du 2).*

Remarque 3.6.3.6. *Tout comme dans la procédure LB-FSDS, la borne inférieure fournie par la procédure LB-WFSDS serait plus judicieusement utilisée dans une procédure de type branch and bound qu'à l'évaluation de la qualité d'une valeur trouvée par une méthode.*

Nous illustrons ces critères de décision avec des tâches pondérées et des temps logistiques dépendant de la séquence des tâches dans l'exemple ci-après.

3.6.4 Illustration de la procédure avec des temps logistiques et des poids

Exemple 3.5

Pour cet exemple 3.5, nous utilisons les caractéristiques des tâches dans l'instance de l'exemple

3.3 et les temps logistiques dans l'exemple 3.4. L'instance de l'exemple 3.5 est donc représentée sur les tableaux 3.20 et 3.21.

i	1	2	3	4	5
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03
Poids (w_i)	0,91	1,46	2,57	2,02	1,84

Table 3.20 – Instance de l'exemple 3.5 pour la fonction PRTWFST

$\downarrow i, j \rightarrow$	1	2	3	4	5
0	0,38	0,70	0,20	2,97	0,71
1	-	0,46	0,27	1,69	0,48
2	0,28	-	0,32	0,81	0,86
3	0,71	1,53	-	0,43	0,19
4	0,53	0,29	1,71	-	1,59
5	0,28	0,04	0,09	0,26	-

Table 3.21 – Matrice des temps logistiques s_{ij} dans l'exemple 3.5

La tâche complétée à l'instant initial est la tâche fictive 0. Les tableaux 3.22, 3.23 et 3.24 donnent respectivement les valeurs de la fonction PRTWFST, la matrice de dominance et le vecteur des forces des tâches à cette date.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	4,48	6,61	8,29	5,42
2	8,95	-	18,61	16,41	15,37
3	12,26	15,77	-	16,37	15,22
4	9,65	11,50	19,34	-	15,33
5	4,05	5,37	8,58	7,37	-

Table 3.22 – Valeurs de la fonction PRTWFST à $t = 0$ dans l'exemple 3.5

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	1	0
2	0	-	0	0	0
3	0	1	-	1	0
4	0	1	0	-	0
5	1	1	1	1	-

Table 3.23 – Matrice de dominance à $t = 0$ dans l'exemple 3.5

Sur la base du vecteur des forces des tâches, la tâche 5 est sélectionnée à l'instant $t = 0$. Les autres décisions sont regroupées dans le tableau 3.25.

La séquence d'exécution des tâches de l'exemple 3.5 est représentée sur le tableau 3.26 avec un temps de réponse pondéré moyen par tâche égal à 5,0951. L'illustration de la procédure la

Tâche i	1	2	3	4	5
Force f_i	3	0	2	1	4

Table 3.24 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 3.5

Instant de décision (t)	2,74	3,50	4,44	5,73
Tâche complétée	5	3	4	1
Ensemble des tâches	{1, 2, 3, 4}	{1, 2, 4}	{1, 2}	{2}
Tâches sélectionnée	3	4	1	2
Date de début (après set-up)	3,28	3,93	4,97	6,19

Table 3.25 – Autres décisions dans l'exemple 3.5 pour la fonction PRTWFSST

borne inférieure LB-WFSDS ne diffère de celle de la procédure LB-WFSDS que par la division de la somme du temps logistique et de la durée d'exécution par le poids. La valeur de cette borne inférieure obtenue pour l'exemple 3.5 est 3,3277 avec une séquence des morceaux de tâches et de temps logistiques représentés sur le tableau 3.27.

Ordre d'exécution	1	2	3	4	5
Tâche	5	3	4	1	2
Date de début (après set-up)	0,71	3,28	3,93	4,97	6,19
Date de complétion	2,74	3,50	4,44	5,73	9,13

Table 3.26 – Séquence d'exécution des tâches de l'exemple 3.5

Ordre d'exécution	1	2	3	4	5	6	7
Tâche	5	1	5	4	4	5	1
Date de début	0	0,19	0,28	0,92	1,18	1,69	3,08
Date de fin/interruption	0,19	0,28	0,92	1,18	1,69	3,08	3,19
Ordre d'exécution	8	9	10	11	12	13	
Tâche	3	3	1	1	2	2	
Date de début	3,19	3,28	3,50	3,58	4,34	4,38	
Date de fin/interruption	3,28	3,50	3,58	4,34	4,38	7,32	

Table 3.27 – Séquence d'exécution des tâches dans la borne inférieure de l'exemple 3.5

Sur la base de tout ce qui précède, nous reconsidérons maintenant problème (\mathcal{P}_{14}) dans sa forme temps-réel. La procédure de décision en temps-réel est l'objet de la sous-section suivante.

3.6.5 Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{14})

Nous utiliserons, en plus de tout ce qui précède, la proposition suivante dans la procédure de décision en temps-réel. Cette proposition nous permettra par la suite, comme dans les problèmes (\mathcal{P}_{11}), (\mathcal{P}_{12}) et (\mathcal{P}_{13}), de parler indifféremment des entités ou des demandes de tâches de maintenance sur ces entités. La justification de cette proposition est analogue à celle de la proposition 3.3.1.3.

Proposition 3.6.5.1. *En respectant le théorème 3.6.1.2 avec la fonction PRTWFSST, on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} .*

En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, le théorème 3.6.1.2 permet de programmer la $m_1^{\text{ième}}$ tâche sur \mathcal{E}_{ik} avant la $m_2^{\text{ième}}$.

Ainsi, dans la suite de cette section, nous parlerons indifféremment de sélection d'une tâche sur une entité que de sélection d'une entité. Dans la procédure temps-réel, nous utiliserons en plus de la fonction PRTWFST, du théorème 3.6.1.2 et de la proposition 3.6.5.1, la définition 3.3.1.4 d'urgence des tâches.

Partant de tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{14}).

Algorithme 3.6.5.2. Procédure en temps-réel OL-MTSA-WST

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques sur chaque entité du système comme définies dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource).
La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.
4. Appliquer la procédure décrite dans l'algorithme AlgoOF-PRTWFST 3.6.2.1 à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} à sa date de fin.
Aller à l'étape 1.
5. Fin.

Proposition 3.6.5.3. Complexité

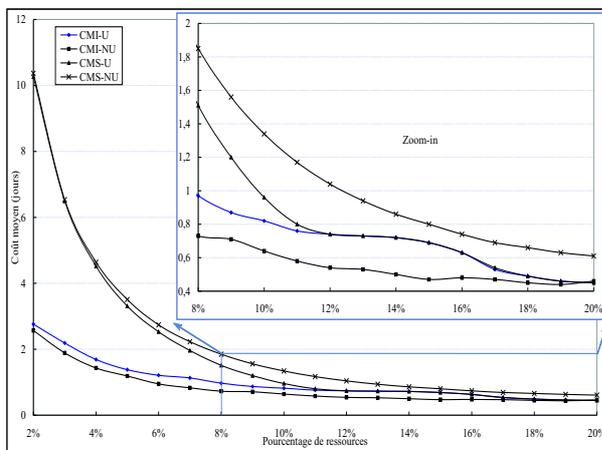
Si N^l est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité des procédures OL-MTSA-WST est en $O\left(N^l \left(\sum_{k=1}^N N_k\right)^2\right)$.

La justification de cette proposition est analogue à celle de la complexité de la procédure OL-MTSA-W dans la proposition 3.4.5.3.

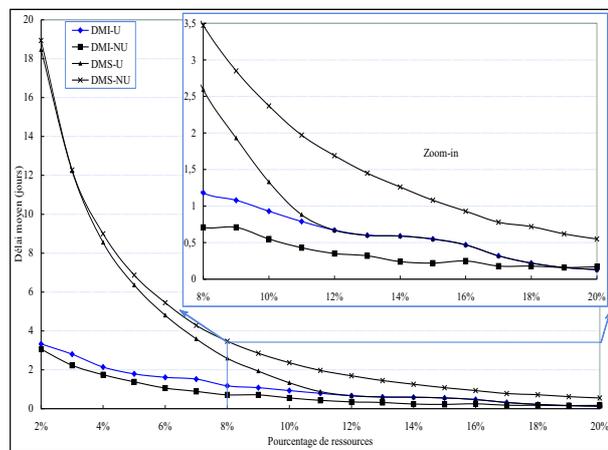
3.6.6 Résultats numériques

Expérimentations

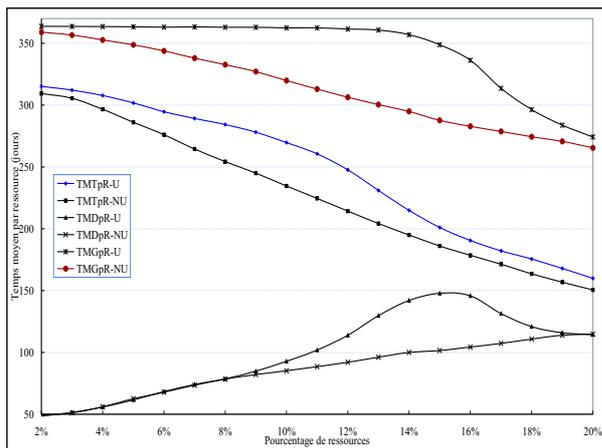
Pour les expérimentations, nous reconsidérons le protocole d'expérimentation décrit dans la sous-section 3.3.3 en considérant à présent les données relatives aux temps logistiques entre les sites, et celles relatives aux poids des entités. Les conditions d'expérimentation ici sont les mêmes que celles de la section précédente, à part les poids des entités qui viennent s'y rajouter. Les résultats sont résumés sur la figure 3.10, avec les mêmes appellations des paramètres que dans la section précédente, tout en sachant qu'ici, les coûts des états critiques CMI et CMS correspondent aux temps de réponses pondérés des tâches de maintenance.



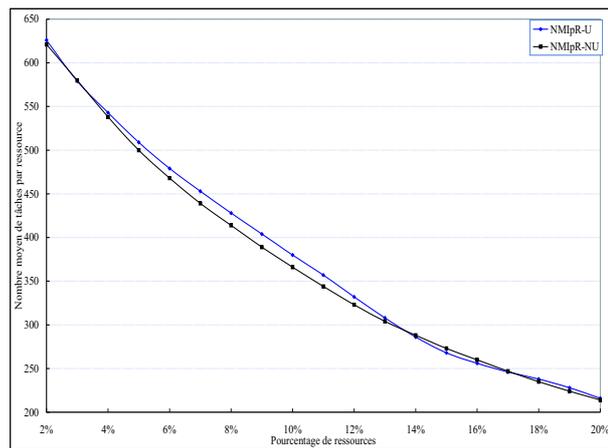
(a) Coût moyen de maintenance (en temps de réponse pondéré)



(b) Délai moyen d'intervention



(c) Temps moyen d'activité par ressource



(d) Nombre moyen de tâches par ressource

Figure 3.10 – Résultats avec des poids inégaux et des temps logistiques importants

Analyse des résultats

Les observations qui découlent des résultats dans cette configuration sont analogues à celles de la sous-section 3.5.6. Pour davantage de détails sur cette configuration, le lecteur peut se référer à [5].

3.7 Conclusion

Dans ce chapitre, nous avons abordé les problèmes de décision en temps-réel dans un processus de maintenance dans un système distribué. Ces systèmes distribués comprennent plusieurs sites, chaque site étant composé d'entités réparables, et il y a généralement des temps logistiques considérables entre ces différents sites. Nous avons considéré ce système distribué sous différentes configurations qui sont des combinaisons des deux critères suivants.

- Les différentes entités du système sont également ou inégalement pondérées. Les poids des entités sont liés à des coûts de maintenance par unité de temps relatifs à chacune de ces entités, coûts qui sont en général en relation avec la situation de ces entités dans le système, par rapport à leur importance pour la réalisation d'une fonction particulière, leur poids économique, leur sensibilité, etc.
- Les temps logistiques entre les différents sites du système sont faibles, de façon qu'on puisse les assimiler à 0 dans l'évolution du système, ou bien trop importants pour qu'on puisse faire d'hypothèses réductrices à ce propos. Il est naturel de considérer les temps logistiques importants dans un système distribué. Mais les différents sites du système pourraient être par exemple regroupés sur un même site géographique, ce qui aurait quelquefois pour conséquence de réduire les temps logistiques. Dans certaines de ces mesures, ces temps logistiques peuvent devenir faibles. C'est dans cette optique que se place l'hypothèse des temps logistiques faibles.

Dans chacune de ces configurations et pour les besoins de la prise de décision en temps-réel dans le processus de maintenance, nous avons considéré des critères de décision localement optimaux sur des problèmes localement statiques. Les développements nous ont conduits à des fonctions pour l'optimalité locale suivant le critère de coût en jeu. Dans certains cas, les fonctions donnant l'optimalité locale sont relatives à une seule des tâches de maintenance à la fois et définissent des relations d'ordre dans l'ensemble de tâches à chaque date de décision. Dans ces cas, leur utilisation est moins difficile. C'est le cas lorsque les entités du système sont également pondérées et que les temps logistiques entre les sites sont faibles. C'est aussi le cas lorsque les entités sont également pondérées et que les temps logistiques entre les sites sont considérables, mais symétriques d'un site à un autre. Dans les autres cas, les fonctions donnant l'optimalité locale ne sont pas relatives à une seule tâche de maintenance à la fois. Dans ces cas, nous avons basé leur utilisation sur deux concepts que nous avons introduits dans ce chapitre. Ces deux concepts sont la matrice de *dominance* des tâches et le vecteur des *forces* des tâches. À l'aide d'exemples illustratifs simples, nous avons montré comment l'usage de ces concepts peut être faite. Ils peuvent même

être facilement utilisés dans la résolution manuelle des problèmes.

Dans tous les cas considérés avec des entités pondérées ou non, avec des temps logistiques considérables ou faibles, les règles de priorités d'optimalité locale sur lesquelles sont basés les algorithmes dans les cas localement statiques, sont suivies de procédures de calcul de bornes inférieures. En particulier, lorsque les temps logistiques sont considérables et dépendent de la séquence d'exécution des tâches, nous avons proposé une procédure de borne inférieure lorsque les tâches sont également pondérées. Cette procédure s'est révélée par la suite être un cas particulier de la procédure de borne inférieure que nous avons proposée lorsque les tâches sont inégalement pondérées. Ainsi, lorsque le problème est un problème statique de minimisation de temps de réponse total (pondéré ou non) sur une ressource ("single machine" en ordonnancement), avec des tâches (pondérées ou non), des temps logistiques dépendant de la séquence des tâches (existant ou non), et des dates d'occurrences (ou encore dates d'arrivée) inégales de tâches, ces fonctions d'optimalité locale, qui sont des propriétés de dominance, et les bornes inférieures peuvent permettre de développer des méthodes exactes du type *branch and bound* pour ces problèmes d'ordonnancement statiques.

Dans tout ce chapitre, les tâches de maintenance sur les entités sont supposées avoir des durées d'exécution fixes connues qui sont leur *MTTR*. Par ailleurs, les événements critiques considérés sur les entités sont réduits au franchissement de seuils critiques sur les disponibilités de ces entités. Le cas le plus présent ici est donc le cas de la maintenance préventive par rapport à l'évolution de la disponibilité des entités du système. Les critères de coût considérés sont principalement liés au temps passé des états critiques du système étudié. Mais il peut s'avérer que les criticités des entités du système ne soient pas les seuls points en vue. D'autres coûts peuvent être liés à d'autres caractéristiques comme, par exemple, des délais à respecter par les interventions de maintenance dans le but de réduire certaines pertes spécifiques. Certains de ces aspects seront considérés dans le chapitre 4 suivant.

Par ailleurs, les procédures décrites dans ce chapitre peuvent être étudiées en simulation dans les cas où les durées des tâches de maintenance sont plutôt aléatoires et où des pannes fortuites interviennent dans le fonctionnement du système. Nous reviendrons là dessus dans le chapitre 5 pour comparer les différents résultats dans ce cas et en même temps évaluer la robustesse de ces approches.

Chapitre 4

Maintenance avec coûts impliquant des contraintes de délais

4.1 Introduction

Dans le chapitre précédent, nous avons proposé des approches pour la prise de décision en temps-réel dans un processus de maintenance sur les systèmes distribués afin de maîtriser des coûts liés notamment à des états critiques de ces systèmes. C'est l'une des principales préoccupations énumérées dans les objectifs de ce travail. En général, les états critiques ne sont pas les seuls points noirs. D'autres préoccupations existent dans des cas comme par exemple lorsqu'il y a des contraintes additionnelles de délais à propos des interventions de maintenance. Ce chapitre est justement consacré à certaines de ces autres préoccupations, notamment celles liées aux retards des tâches de maintenance en plus des états critiques ou à des coûts de fréquences élevées ou de déficit de maintenance. Les coûts combinés des états critiques et des retards des tâches de maintenance se rapportent à des critères combinant les temps de réponse et les retards de tâches de maintenance, alors que les coûts de sur-maintenance et de déficit de maintenance se rapportent à des critères combinant les avances et les retards de tâches de maintenance. Tout ceci sera expliqué de façon détaillée dans la position du problème.

Ce chapitre est composé comme suit. Dans la section 2, les deux types de préoccupation ci-dessus seront spécifiés en détail dans la position du problème. Ensuite, chacune de ces préoccupations sera considérée suivant la configuration du système distribué : lorsque les temps logistiques sont faibles et lorsqu'ils sont non négligeables. Ainsi, dans la section 3, nous allons proposer des approches pour la prise de décision en temps-réel pour le problème combiné des coûts d'états critiques de retard des tâches de maintenance mais avec des temps logistiques faibles entre les sites alors que ce même problème avec des temps logistiques considérables sera pris en compte dans la section 4. Après cela, le problème des coûts dus à des avances et des retards des tâches de maintenance sera traité dans la section 5 avec des temps logistiques faibles entre les sites et dans la section 6 avec des temps logistiques considérables entre les sites. Les principales conclusions de ce chapitre seront tirées dans la section 7. Comme dans le chapitre précédent, les approches de

prise de décision seront illustrées sur des exemples simples avant d'être utilisées sur des systèmes de taille réelle et dans des conditions de fonctionnement réelles.

4.2 Position du problème

Le problème abordé dans ce chapitre se situe dans le même cadre que celui du chapitre 3 mais avec une contrainte additionnelle de délai sur les tâches de maintenance. C'est un problème de la recherche de techniques appropriées pour l'allocation en temps-réel des ressources pour la réalisation des opérations de maintenance d'un système industriel distribué. L'étude porte sur un système formé d'un nombre donné de sites différents, chaque site étant composé d'entités réparables. Le système dispose de ressources partagées par l'ensemble des sites. Le processus de maintenance est abordé, tout comme dans le chapitre précédent, sur un horizon fini \mathcal{H} . Les sites du système fonctionnent de façon indépendante ; les entités d'un site donné fonctionnent en série, ce qui implique l'obligation de disponibilité de chaque entité d'un site donné pour le bon fonctionnement de celui-ci.

Les notations utilisées dans le chapitre 3 précédent sont toujours valables dans ce chapitre-ci. En plus de ces notations qui sont N , M , S_k , N_k , \mathcal{E}_{ik} , λ_{ik} , μ_{ik} , p_{ik} , $A_{ik}(t)$, \mathcal{H} , $C_{tot}(\mathcal{H})$, α_{ik}^1 , β_{ik} , γ_{ik} , σ_{ik} , τ_{ik} , $r_{m,ik}$, $c_{m,ik}$, w_{ik} et s_{kl} , nous introduirons quelques notations supplémentaires dans ce chapitre telles que α_{ik}^2 , α_{ik}^3 , τ_{ik}^2 , τ_{ik}^3 , $d_{m,ik}$, W_e , W_f , W_t , $C_2(\mathcal{H})$, $C_2^w(\mathcal{H})$, $C_3(\mathcal{H})$ et $C_3^w(\mathcal{H})$. De même que dans le chapitre précédent, les tâches de maintenance (préventive et curative) sont supposées être exécutées sans préemption. Les dates de remise en service d'une entité seront confondues avec les dates de complétion des tâches de maintenance. Les durées des tâches de maintenance sur les entités sont supposées constantes sur l'horizon de planification \mathcal{H} et égales à leurs *MTTR* et chaque intervention (tâche de maintenance) sur une entité renouvelle l'état de celle-ci. Par ailleurs, tout comme dans le chapitre précédent, le nombre M de ressources est très inférieur au nombre total d'entités $\sum_{k=1}^N N_k$ dans le système ($M \ll \sum_{i=1}^N N_k$). Souvent, deux autres types de problèmes se posent et impliquent des contraintes de délais sur les tâches de maintenance. Nous posons ces deux types de problèmes dans les deux sous-sections qui suivent.

4.2.1 Problème avec des coûts d'états critiques et de retards

Dans le chapitre 3 précédent, il était question qu'une entité \mathcal{E}_{ik} tombe en état critique lorsque sa disponibilité A_{ik} descend en dessous d'un seuil critique α_{ik}^1 . En général, un autre type de problème se pose. Une entité en état critique, tant qu'elle n'est pas en panne, peut continuer de fonctionner avec, bien sûr, une dégradation de plus en plus poussée. Au delà d'un certain stade de dégradation, il n'y a plus que des coûts d'états critiques engendrés par le système. Il y a des coûts supplémentaires dus notamment à des pertes excessives de production et à des actions supplémentaires de maintenance (actions palliatives par exemple), etc. Ainsi, si nous reprenons le raisonnement analogue à celui du chapitre 3, une entité \mathcal{E}_{ik} est en état critique si sa disponibilité descend en dessous du seuil critique α_{ik}^1 . En dessous d'un deuxième seuil de disponibilité α_{ik}^2 (avec $0 < \alpha_{ik}^2 < \alpha_{ik}^1 < 1$), elle engendre un coût additionnel de retard de maintenance.

Ainsi, pour une entité \mathcal{E}_{ik} qui a un comportement exponentiel, avec une disponibilité A_{ik} à l'instant t renouvelée par une remise en service l'instant T

$$A_{ik}(t - T) = \frac{\mu_{ik}}{\lambda_{ik} + \mu_{ik}} + \frac{\lambda_{ik}}{\lambda_{ik} + \mu_{ik}} \exp[-(\lambda_{ik} + \mu_{ik})(t - T)] \quad (4.2.1)$$

elle tombe en état critique après une durée τ_{ik} (relative au premier seuil α_{ik}^1). Un délai est alors imposé sur la tâche de maintenance qui se situe à une durée notée τ_{ik}^2 (relative au deuxième seuil α_{ik}^2) de sa dernière remise en service, avec $\tau_{ik}^2 > \tau_{ik}$. Les durées τ_{ik} et τ_{ik}^2 sont respectivement définies, de la même façon que dans le chapitre 3, par

$$\tau_{ik} = \frac{-1}{\lambda_{ik} + \mu_{ik}} \ln \left[\alpha_{ik}^1 \left(1 + \frac{\mu_{ik}}{\lambda_{ik}} \right) - \frac{\mu_{ik}}{\lambda_{ik}} \right] \quad (4.2.2)$$

et

$$\tau_{ik}^2 = \frac{-1}{\lambda_{ik} + \mu_{ik}} \ln \left[\alpha_{ik}^2 \left(1 + \frac{\mu_{ik}}{\lambda_{ik}} \right) - \frac{\mu_{ik}}{\lambda_{ik}} \right] \quad (4.2.3)$$

Dans le cas la loi de Weibull, avec un taux de réparation μ_{ik} constant sur l'horizon de planification \mathcal{H} pour une entité \mathcal{E}_{ik} , nous rappelons que la disponibilité à l'instant t , après une mise ou une remise en service à l'instant T est $A_{ik}(t - T)$ où $A_{ik}(t)$ s'exprime de la même façon que $A(t)$ ci-après.

$$A(t) = \exp \left(- \left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \mu \sigma \cdot \int_0^{\frac{t - \gamma}{\sigma}} \exp(\mu \sigma x + x^\beta) dx \right\} \quad (4.2.4)$$

Nous rappelons aussi que lorsque le paramètre de forme β_{ik} est supérieur ou égal à 1 ($\beta_{ik} \geq 1$, correspondant respectivement à la phase de vie stable de l'entité et à sa phase de vieillesse), la fonction disponibilité $A_{ik}(t)$ est strictement décroissante et tend vers une limite $A_{ik}(\infty)$ telle que $\left(0 \leq A_{ik}(\infty) < \frac{\mu_{ik}}{\mu_{ik} + \beta_{ik}/\sigma_{ik}} \right)$. Ainsi, pour une telle entité \mathcal{E}_{ik} , il existe une unique durée τ_{ik} telle que l'on ait l'égalité $A_{ik}(\tau_{ik}) = \alpha_{ik}^1$ et une unique durée τ_{ik}^2 telle que l'on ait l'égalité $A_{ik}(\tau_{ik}^2) = \alpha_{ik}^2$. τ_{ik} et τ_{ik}^2 peuvent être simulées à partir de la fonction disponibilité de l'entité \mathcal{E}_{ik} .

Ainsi, si $c_{m,ik}$ désigne la date de la m^{ieme} remise en service (confondue avec la date de complétion de la m^{ieme} tâche de maintenance sur l'entité \mathcal{E}_{ik}), il y aura occurrence d'état critique à la date $r_{m+1,ik}$ et un délai sur la tâche de maintenance à la date $d_{m+1,ik}$, avec

$$r_{m+1,ik} = c_{m,ik} + \tau_{ik} \text{ et } d_{m+1,ik} = c_{m,ik} + \tau_{ik}^2 \quad (4.2.5)$$

Le coût total des états critiques et des retards de maintenance sur l'horizon de planification \mathcal{H} est alors

$$C_2(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} ((c_{m,ik} - r_{m,ik}) + \max(0, c_{m,ik} - d_{m,ik})) \right) \quad (4.2.6)$$

Remarque 4.2.1.1. Si la situation est telle que les coûts d'états critiques et de retard n'ont pas la même importance pour le système, ils peuvent être inégalement pondérés respectivement par W_f et W_t . Dans ce cas, le coût total des états critiques et des retards sur l'horizon de planification sera

$$C_2^w(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} (W_f \cdot (c_{m,ik} - r_{m,ik}) + W_t \cdot \max(0, c_{m,ik} - d_{m,ik})) \right) \quad (4.2.7)$$

Chaque intervention (tâche de maintenance) sur l'entité \mathcal{E}_{ik} a une durée p_{ik} qui est supposée,

dans un premier temps, être constante sur l'horizon de planification \mathcal{H} et égale à son *MTTR* $\left(\frac{1}{\mu_{ik}}\right)$. Le problème peut alors être formulé comme suit :

(\mathcal{P}_2) : *Minimiser le coût $C_2(\mathcal{H})$ (ou $C_2^w(\mathcal{H})$) (le coût des états critiques et des retards de maintenance sur l'horizon \mathcal{H}) avec M ressources parallèles à performances identiques, partagées par toutes les composantes du système, sachant que l'occurrence des événements intervient à tout moment sur l'horizon, avec des temps logistiques entre les différents sites du système, tout en gardant au mieux les conditions $A_{ik} \geq \alpha_{ik}^1$.*

Ce problème, tout comme le problème (\mathcal{P}_1) , est un problème d'organisation en temps-réel. Le temps logistique est s_{kl} entre deux sites \mathcal{S}_k et \mathcal{S}_l et ces temps logistiques dépendent de la séquence d'exécution des tâches de maintenance. Les ressources ont des performances identiques et le problème est un problème à ressources parallèles identiques. Sa résolution se fera en deux étapes, suivant les configurations du système à savoir :

- Les temps logistiques entre les sites sont faibles ; nous l'appellerons problème (\mathcal{P}_{21}) .
- Les temps logistiques entre les sites sont considérables ; nous l'appellerons problème (\mathcal{P}_{22}) .

Avec un raisonnement analogue au chapitre précédent, le problème (\mathcal{P}_2) a la propriété suivante dans les deux cas de figure.

Proposition 4.2.1.2. *Le problème (\mathcal{P}_2) est NP-difficile dans les deux cas de figure ci-dessus.*

Les procédures de résolution se feront dans les sections qui vont suivre, mais avant d'aborder la résolution, introduisons le deuxième type de problème qui doit faire face à des contraintes de délais dans la gestion en temps-réel de la maintenance.

4.2.2 Problème avec des coûts d'avances et retards

Outre le problème impliquant la combinaison des coûts des états critiques des entités et de retard des tâches de maintenance, un autre problème impliquant des contraintes de délais dans la gestion en temps-réel de la maintenance est le problème relatif à la question de maintenance juste à temps (*JIT* - Just In Time, en gestion de production). En général, il est souhaitable d'exécuter les interventions de maintenance à des dates bien données. Compte tenu des caractéristiques des systèmes, une maintenance préventive entreprise avant ces dates est une maintenance préventive entreprise trop souvent, c'est-à-dire de la sur-maintenance. En revanche, passé ces dates c'est plutôt de la maintenance entreprise pas assez souvent, autrement dit un déficit de maintenance.

Pour modéliser cet aspect du problème, nous considérons toujours le système avec des sites indépendants composés d'entités réparables et disposant des ressources communes pour la maintenance sur tous les sites. Pour une entité \mathcal{E}_{ik} avec une fonction disponibilité A_{ik} , un seuil α_{ik}^3 représente le stade pour une maintenance idéale. Une maintenance préventive entreprise sur l'entité \mathcal{E}_{ik} avant le seuil α_{ik}^3 engendre un coût d'avance de maintenance. En revanche, une maintenance préventive non accomplie au delà de ce seuil engendre un coût de retard de maintenance. Un tel problème implique des pénalités d'avance et de retard (E/T - Earliness/Tardiness), mais

tout comme les précédents, c'est aussi un problème de gestion en temps-réel. Ainsi, une entité \mathcal{E}_{ik} mise en service ou remise en service à une date T , atteindra le seuil requis α_{ik}^3 après une durée de τ_{ik}^3 . Les dates relatives à ces durées seront aussi, sans risque de confusion avec le cas de la sous-section précédente, des délais ($d_{m,ik}$ pour la m^{ieme} remise en service). Si l'entité \mathcal{E}_{ik} a une disponibilité exponentielle, la durée τ_{ik}^3 sera obtenue, de façon analogue aux précédentes 4.2.2 et 4.2.3, par l'expression 4.2.1 de la fonction disponibilité,

$$\tau_{ik}^3 = \frac{-1}{\lambda_{ik} + \mu_{ik}} \ln \left[\alpha_{ik}^3 \left(1 + \frac{\mu_{ik}}{\lambda_{ik}} \right) - \frac{\mu_{ik}}{\lambda_{ik}} \right] \quad (4.2.8)$$

Pour une entité suivant une loi de Weibull, la durée τ_{ik}^3 sera simulée de la même façon que les cas précédents à partir de la fonction disponibilité A_{ik} dans l'expression 4.2.4 et du seuil α_{ik}^3 .

Le coût globale des avances et des retards de maintenance sur l'horizon de planification \mathcal{H} sera alors

$$C_3(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/d_{m,ik} \in \mathcal{H}} (\max(0, c_{m,ik} - d_{m,ik}) + \max(0, d_{m,ik} - c_{m,ik})) \right) \quad (4.2.9)$$

soit

$$C_3(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/d_{m,ik} \in \mathcal{H}} |c_{m,ik} - d_{m,ik}| \right) \quad (4.2.10)$$

Remarque 4.2.2.1. *La situation peut être telle que les coûts d'avance et de retard n'aient pas la même importance pour le système, ils peuvent alors être inégalement pondérés respectivement par W_e et W_t . Dans ce cas, le coût total des avances et retards de maintenance sur l'horizon de planification \mathcal{H} sera*

$$C_3^w(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/d_{m,ik} \in \mathcal{H}} (W_e \cdot \max(0, c_{m,ik} - d_{m,ik}) + W_t \cdot \max(0, d_{m,ik} - c_{m,ik})) \right) \quad (4.2.11)$$

Le problème peut alors être formulé comme suit :

(\mathcal{P}_3) : *Minimiser le coût $C_3(\mathcal{H})$ (ou $C_3^w(\mathcal{H})$) (le coût des avances et des retards de maintenance sur l'horizon \mathcal{H}) avec M ressources parallèles à performances identiques, partagées par toutes les composantes du système, sachant que l'occurrence des événements intervient à tout moment sur l'horizon, avec des temps logistiques entre les différents sites du système.*

Nous reviendrons sur ce problème plus tard dans des cas où des fenêtres temporelles sont imposées pour les interventions de maintenance préventive. Mais dans ce chapitre, nous considérons que, compte tenu des contraintes de production, une entité \mathcal{E}_{ik} remise en service à la date T pour la $(m-1)^{ieme}$ fois ne peut subir d'intervention de maintenance préventive avant une date $r_{m,ik}$, liée à une durée τ_{ik} . Nous traiterons ce problème, comme le précédent, suivant deux configurations à savoir :

- les temps logistiques entre les sites sont faibles ; nous l'appellerons problème (\mathcal{P}_{31}),
- les temps logistiques entre les sites sont importants ; nous l'appellerons problème (\mathcal{P}_{32}).

Il faut noter que dans la présente formulation, il s'agit de dates requises pour des interventions

optimales de maintenance préventive. Mais au lieu de dates optimales, il pourrait s'agir de fenêtres de temps optimales pour des interventions de maintenance préventive. Ce serait alors un problème d'avance et retard de maintenance avec fenêtres horaires. Mais ici, nous considérons plutôt des cas où il s'agit de dates. Une extension de la procédure obtenue pourra se faire pour le cas des fenêtres horaires.

Proposition 4.2.2.2. *Le problème (\mathcal{P}_3) (aussi bien avec le coût $C_3(\mathcal{H})$ que $C_3^w(\mathcal{H})$) est NP-difficile dans les deux cas de figure ci-dessus.*

En effet prenons le problème (\mathcal{P}_3) avec le coût $C_3^w(\mathcal{H})$ dans sa forme la plus simple (\mathcal{P}_{31}), dans une situation statique (où toutes les informations sur les tâches sont connues dès le début du processus) et avec une ressource unique. Lorsque $W_e = 0$ et $W_t = 1$, cela revient en ordonnancement de la production à un problème d'ordonnement de tâches sur machine unique pour la minimisation du retard total, avec des dates d'occurrence ou dates d'arrivée inégales des tâches ($n/1/r_i/\sum \max(0, C_i - d_i)$). Ce problème est NP-difficile (voir par exemple dans [43] et [45]). Ainsi, le problème (\mathcal{P}_{31}) avec le coût $C_3^w(\mathcal{H})$ est NP-difficile.

Pour le problème localement statique avec le coût $C_3(\mathcal{H})$, il est aussi NP-difficile (voir par exemple dans [71]). Le problème (\mathcal{P}_3) est donc NP-difficile avec les deux types de coûts et sous toutes ses formes.

Sous une version pondérée, lorsque toutes les tâches ont une même date d'occurrence, Ow et Morton [123], d'une part, et Liaw [103], d'autre part, ont développé deux règles de dominance qui ont été adaptées par Valente et Alves [157], [158] dans une heuristique et une branch and bound pour le problème statique avec des dates d'occurrence inégales des tâches. La procédure que nous proposerons pour le problème en temps-réel reposera sur une relation d'optimalité locale prenant aussi en compte les dates d'occurrence inégales des tâches de maintenance.

Maintenant, nous considérons les approches de résolution des problèmes (\mathcal{P}_2) et (\mathcal{P}_3). La section suivante considère le problème (\mathcal{P}_2) sous sa configuration (\mathcal{P}_{21}) où les temps logistiques entre les sites sont faibles.

4.3 Coûts d'états critiques et retards avec des temps logistiques faibles

Nous abordons dans cette section le problème (\mathcal{P}_{21}) qui est la première configuration du problème (\mathcal{P}_2). Il s'agit de résoudre le problème d'organisation en temps-réel de la maintenance sur le système distribué spécifié dans la position du problème afin de minimiser, sur un horizon fini \mathcal{H} , le coût $C_2(\mathcal{H})$ (resp. $C_2^w(\mathcal{H})$) des états critiques des entités du système et des retards des tâches de maintenance tel qu'exprimé dans l'expression 4.2.6 (resp. 4.2.7) et ce, avec une configuration du système telle que les temps logistiques entre les différents sites soient faibles et assimilés à 0. Ce coût est une combinaison de deux coûts : le coût des temps de réponse pour les tâches de maintenance (si nous reprenons le terme utilisé dans le chapitre précédent) et celui dû à leurs retards. Ces deux coûts sont normalement à minimiser chacun de son côté. Cependant dans la théorie de l'optimisation multi-critères, il est souvent procédé, dans de pareilles situations, à la minimisation

de l'un des coûts sous la contrainte que le second soit inférieur à une borne fixée. Il est aussi procédé à une optimisation du type Paréto. On considère encore une combinaison des différents coûts à optimiser. C'est dans ce troisième cas de figure que se retrouvent les coûts $C_2(\mathcal{H})$ et $C_2^w(\mathcal{H})$.

Le problème (\mathcal{P}_{21}) , comme nous l'avons dit plus haut, est NP-difficile. C'est aussi un problème de gestion en temps-réel. Comme dans le chapitre 3, les décisions se feront localement sur des problèmes localement statiques, avec des options de décision localement optimales. Les développements ultérieurs se feront principalement sur la base du coût $C_2(\mathcal{H})$. Nous allons donc considérer, dans le problème statique, un ensemble de n tâches (disponible ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Les développements sur ce problème localement statique sont l'objet de la sous-section ci-après.

4.3.1 Le problème statique et les règles de décision

Soit donc un ensemble de n tâches (disponible ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Chaque tâche i a une date d'occurrence r_i (qui diffère d'une tâche à une autre), une durée d'exécution p_i et un délai d_i . Les tâches sont exécutées sans préemption. L'objectif est de minimiser leur coût combiné de temps de réponse et de retard par rapport aux délais d_i ($\sum_{i=1}^n ((c_i - r_i) + \max(0, c_i - d_i))$), ou sous forme pondérée ($\sum_{i=1}^n (W_f \cdot (c_i - r_i) + W_t \cdot \max(0, c_i - d_i))$) où c_i est la date de complétion de la tâche i . Dans l'optique de la résolution de ce problème localement statique, posons la définition suivante.

Définition 4.3.1.1. À tout instant t , la fonction *FTR* (*Flow-time and Tardiness Rule*) est définie pour un couple de tâches (i, j) comme suit :

$$FTR(i, j, t) = \max[PRTF(i, t), Q_1(i, j, t)] + \max[PRTT(i, t), Q_2(i, j, t)] \quad (4.3.1)$$

avec

$$PRTF(i, t) = 2\max(r_i, t) + p_i,$$

$$PRTT(i, t) = \max(r_i, t) + \max(\max(r_i, t) + p_i, d_i),$$

$$Q_1(i, j, t) = \max(r_i, t) + \max(r_j, t)$$

$$Q_2(i, j, t) = \max(\max(r_i, t), d_i - p_i) + \max(\max(r_j, t), d_j - p_j).$$

Dans ce chapitre, nous utiliserons encore la relation de dominance d'une tâche sur une autre et la notion de force des tâches telles que définies dans le chapitre 3. Avec cette définition de la fonction *FTR*, le théorème ci-après est vérifié.

Théorème 4.3.1.2. Soit une paire de tâches $\{i, j\}$ à ordonnancer à la fin d'une séquence partielle de tâches π , par rapport au coût ($\sum_{i=1}^n ((c_i - r_i) + \max(0, c_i - d_i))$). À tout instant t , $i \preceq j$ si et seulement si

$$FTR(i, j, t) \leq FTR(j, i, t) \quad (4.3.2)$$

Preuve. La tâche i a les paramètres r_i , p_i et d_i . La tâche j a les paramètres r_j , p_j et d_j .

En ordonnanciant les tâches i et j à l'instant t à la fin de la séquence partielle π , notons C_{ij} le coût additionnel engendré en ordonnanciant i directement avant j et C_{ji} celui en ordonnanciant

directement j avant i . Alors si on pose $R_i = \max(r_i, t)$ et $R_j = \max(r_j, t)$, on a

$$C_{ij} = \{[R_i + p_i - r_i] + [\max(R_i + p_i - d_i, 0)]\} + \{[\max(R_i + p_i, R_j) + p_j - r_j] + [\max(\max(R_i + p_i, R_j) + p_j - d_j, 0)]\}$$

et

$$C_{ji} = \{[R_j + p_j - r_j] + [\max(R_j + p_j - d_j, 0)]\} + \{[\max(R_j + p_j, R_i) + p_i - r_i] + [\max(\max(R_j + p_j, R_i) + p_i - d_i, 0)]\}.$$

Ainsi,

$$\begin{aligned} C_{ij} - C_{ji} &= \{[R_i + p_i - r_i] + [\max(R_i + p_i - d_i, 0)]\} + \{[\max(R_i + p_i, R_j) + p_j - r_j] + [\max(\max(R_i + p_i, R_j) + p_j - d_j, 0)]\} \\ &\quad - \{[R_j + p_j - r_j] + [\max(R_j + p_j - d_j, 0)]\} - \{[\max(R_j + p_j, R_i) + p_i - r_i] + [\max(\max(R_j + p_j, R_i) + p_i - d_i, 0)]\} \\ &= \{[R_i + p_i - r_i] + [\max(R_i + p_i, R_j) + p_j - r_j]\} + \{[\max(R_i + p_i - d_i, 0)] + [\max(\max(R_i + p_i, R_j) + p_j - d_j, 0)]\} \\ &\quad - \{[R_j + p_j - r_j] + [\max(R_j + p_j, R_i) + p_i - r_i]\} - \{[\max(R_j + p_j - d_j, 0)] + [\max(\max(R_j + p_j, R_i) + p_i - d_i, 0)]\}. \end{aligned}$$

Or, en introduisant la fonction PRTF dans [42], Chu a montré (avec l'expression ci-dessus de Q_1) que

$$[R_i + p_i - r_i] + [\max(R_i + p_i - d_i, 0)] = \max(PRTF(i, t), Q_1(i, j, t)) + (p_i + p_j) - (r_i + r_j)$$

et avec la fonction PRTT dans [43] il a montré que

$$\max(R_i + p_i - d_i, 0) + \max(\max(R_i + p_i, R_j) + p_j - d_j, 0) = \max(Q_{ij}, PRTT(i, t) + (p_i + p_j)) - (d_i + d_j)$$

où

$$\begin{aligned} Q_{ij} &= \max(R_i + p_i, d_i) + \max(R_j + p_j, d_j) \\ &= \max(R_i, d_i - p_i) + \max(R_j, d_j - p_j) + (p_i + p_j) \\ &= Q_2(i, j, t) + (p_i + p_j). \end{aligned}$$

avec l'expression ci-dessus de Q_2 . Ainsi, on a

$$\max(R_i + p_i - d_i, 0) + \max(\max(R_i + p_i, R_j) + p_j - d_j, 0) = \max(Q_2(i, j, t), PRTT(i, t) + (p_i + p_j)) - (d_i + d_j).$$

En utilisant ces différentes égalités, on a donc

$$\begin{aligned} C_{ij} - C_{ji} &= \max(PRTF(i, t), Q_1(i, j, t)) + \max(Q_2(i, j, t), PRTT(i, t)) - \max(PRTF(j, t), Q_1(i, j, t)) \\ &\quad - \max(Q_2(i, j, t), PRTT(i, t)) \\ &= FTR(i, j, t) - FTR(j, i, t) \end{aligned}$$

D'où $i \preceq j$ si et seulement si $FTR(i, j, t) \leq FTR(j, i, t)$. □

Remarque 4.3.1.3. *La fonction FTR définit une règle de priorité d'optimalité locale pour le coût exprimé comme somme du temps de réponse total et du retard total. Cette règle ne définit pas une relation d'ordre total dans l'ensemble des tâches à une date de décision, mais elle peut être utilisée avec la matrice de dominance et le vecteur des forces définis dans le chapitre précédent.*

Remarque 4.3.1.4. *Il faut noter que si le coût est exprimé comme une somme pondérée du temps de réponse et du retard, avec le temps de réponse pondéré par W_f et le retard pondéré par W_t ,*

alors le théorème 4.3.1.2 est toujours valable en remplaçant la fonction FTR par celle qui suit

$$FTRW(i,j,t) = W_f \cdot \max(PRTF(i,t), Q_1(i,j,t)) + W_r \cdot \max(PRTT(i,t), Q_2(i,j,t)) \quad (4.3.3)$$

Ainsi, tous les développements ultérieurs sont toujours valables avec cette nouvelle fonction.

Le principe de base de l'utilisation des fonctions FTR et $FTRW$, du théorème 4.3.1.2 et des concepts antérieurement définis, est dans la sous-section suivante.

4.3.2 Algorithme de base

Dans cet algorithme de base, nous considérons un ensemble de n tâches dont on connaît les caractéristiques. Chaque tâche i a une date d'occurrence r_i , une durée d'exécution p_i et un délai d_i . L'objectif de leur ordonnancement est de minimiser la combinaison linéaire de leur temps de réponse et de leur retard. La procédure ci-dessous est destinée au problème statique ou localement statique.

Algorithme 4.3.2.1. Principe de base du FTR et du $FTRW$: AlgoOF-FTR, AlgoOF-FTRW Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée, les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$.
 2. Calculer $FTR(i,j,t)$ (ou $FTRW(i,j,t)$ selon le cas) pour tous les couples de tâches (i,j) dans S .
 3. Calculer la matrice de dominance Ω dans S .
 4. Calculer le vecteur des forces (f_i) dans S .
 5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
- Tant que $\text{card}(S) > 1$, aller à l'étape 4.*

Dans cette procédure, la matrice des valeurs de la fonction FTR (resp. $FTRW$) et la matrice de dominance sont calculées une seule fois à chaque date de décision. Quand on réduit l'ensemble de recherche de la tâche à ordonnancer, les restrictions de ces matrices à ce nouvel ensemble sont considérées. Seul le vecteur des forces des tâches change lorsqu'on change d'ensemble de recherche.

Remarque 4.3.2.2. *Comme les autres algorithmes, l'algorithme 4.3.2.1 peut aussi être adapté dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égale à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant le plus petit indice parmi elles. Le choix des tâches se fait suivant les mêmes étapes que précédemment.*

La proposition suivante donne la complexité de l'algorithme AlgoOF-FTR (resp. AlgoOF-FTRW). La démonstration de cette proposition est analogue est celle de l'Algorithme AlgoOF-PRTWF dans la proposition 3.4.3.3 du chapitre 3.

Proposition 4.3.2.3. *L'algorithme AlgoOF-FTR (resp. AlgoOF-FTRW) basé sur le principe de la fonction FTR (resp. FTRW), de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$ si n est le nombre de tâches à ordonnancer.*

Avant de passer à des exemples simples d'illustration de ces procédures de base, puis à son utilisation dans la procédure de décision en temps-réel dans le processus de maintenance, une borne inférieure du coût exprimé en fonction du temps de réponse et du retard des tâches de maintenance (dans la procédure statique et sur une machine unique) est donnée ci-après.

4.3.3 Borne inférieure du coût de temps de réponse et de retard dans le cas statique

La borne inférieure est évaluée sur une ressource avec la préemption permise des tâches. Elle utilise la règle SPRT (Shrotest Remaining Processing Time) et une borne inférieure développée par Chu dans [43] pour les pénalités de retard en présence de date d'occurrence inégales des tâches, et qui utilise des délais modifiés. La borne inférieure (LB-FTP) est obtenue suivant les étapes qui suivent.

Résumé de la borne inférieure LB-FTP

1. Déterminer la suite $(d'_1, d'_2, \dots, d'_n)$ des délais des tâches triés dans l'ordre croissant (délais modifiés).
2. Ordonnancer les tâches, avec préemption autorisée, suivant la règle SRPT.

Alors, les quantités

$$\sum_{i=1}^n (C_{[i]} - r_{[i]}) + \sum_{i=1}^n \max(C_{[i]} - d'_i, 0)$$

et

$$W_f \cdot \sum_{i=1}^n (C_{[i]} - r_{[i]}) + W_t \cdot \sum_{i=1}^n \max(C_{[i]} - d'_i, 0)$$

où $[i]$ est la tâche complétée à la i^{eme} position et $C_{[i]}$ sa date de complétion, sont des bornes inférieures des coûts exprimés respectivement comme somme et somme pondérée du coût dû aux temps de réponse et aux retards. Ceci est dû au fait que la règle SRPT fournit une borne inférieure du temps de réponse en présence de date d'occurrence inégales des tâches et qu'en plus, comme l'a montré Chu dans [43], les délais modifiés combinés à la règle SRPT donnent une borne inférieure du coût de retard en présence de dates d'occurrences inégales des tâches.

4.3.4 Exemple 4.1 : illustration simple de la procédure AlgoOF-FTR

Nous n'illustrons que la procédure avec la fonction FTR (donc dans les cas où $W_f = W_t = 1$). Lorsque ce n'est pas le cas, l'illustration de la procédure avec la fonction pondérée FTRW se fait de la même façon.

Pour l'illustration de cette procédure, nous reprenons l'instance dans l'exemple 3.1 du chapitre 3. Et dans cette instance, les poids des tâches sont remplacés par leur délais. L'instance de l'exemple 3.1 est ainsi représenté sur le tableau 4.1 ci-après.

i	1	2	3	4	5
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03
Délai (d_i)	1,98	5,78	3,52	1,74	2,51

Table 4.1 – Instance de l'exemple 4.1 pour les fonctions FTR et FTRW

À la date $t = 0$, la matrice des valeurs de la fonction FTR est dans le tableau 4.2.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	7,74	8,94	6,16	6,16
2	16,34	-	16,34	16,34	16,34
3	13,58	13,58	-	13,58	13,58
4	5,79	7,79	8,99	-	5,79
5	5,38	6,14	7,34	5,38	-

Table 4.2 – Valeurs de la fonction FTR à $t = 0$ dans l'exemple 4.1

La matrice de dominance et le vecteur des forces qui s'en suivent sont alors dans les tableaux 4.3 et 4.4 respectivement.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	0	0
2	0	-	0	0	0
3	0	1	-	0	0
4	1	1	1	-	0
5	1	1	1	1	-

Table 4.3 – Matrice de dominance à $t = 0$ dans l'exemple 4.1

Tâche i	1	2	3	4	5
Force f_i	2	0	1	3	4

Table 4.4 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.1

La tâche 5 est alors sélectionnée. Le lecteur peut procéder de la même manière en laissant à chaque fois les tâches déjà ordonnancées jusqu'à ce que la dernière tâche soit ordonnancée. La séquence d'exécution des tâches est alors représentée sur le tableau 4.5, avec un coût combiné

moyen de temps de réponse et de retard égal à 2,95. La borne inférieure de cet exemple fournie par la procédure LB-FTP est 2,42.

Ordre d'exécution	1	2	3	4	5
Tâche	5	4	1	3	2
Date de début	0,28	2,31	2,82	3,58	3,80
Date de fin	2,31	2,82	3,58	3,80	6,74

Table 4.5 – Séquence d'exécution des tâches de l'exemple 4.1

Après ces options de prise de décision dans la problème localement statique avec des temps logistiques faibles entre les sites, nous reconsidérons maintenant problème (\mathcal{P}_{21}) dans sa forme temps-réel.

4.3.5 Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{21})

Dans la procédure de prise de décision en temps-réel, nous utiliserons les différents développements ci-dessus. En outre, la proposition suivante nous permettra par la suite de parler indifféremment, tout comme dans les précédents problèmes, des entités que des demandes de tâches de maintenance sur ces entités.

Proposition 4.3.5.1. *En respectant le théorème 4.3.1.2 avec la fonction FTR ou avec la fonction $FTRW$, on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} . En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, le théorème 4.3.1.2 permet de programmer la $m_1^{ième}$ tâche sur \mathcal{E}_{ik} avant la $m_2^{ième}$.*

La justification de cette proposition est analogue à celle de la proposition 3.3.1.3 dans le chapitre 3. Il suffira pour cela de considérer une entité quelconque \mathcal{E}_{ik} , deux entiers m_1 et m_2 tels que $m_1 < m_2$ (avec deux tâches sur \mathcal{E}_{ik} supposées avoir la même durée d'exécution), de considérer que les dates d'occurrence des tâches et les délais sont telles que $r_{m_1,ik} < r_{m_2,ik}$, $d_{m_1,ik} < d_{m_2,ik}$ et de comparer les valeurs de la fonctions FTR .

Nous parlerons donc indifféremment dans la suite de cette section, de sélection d'une tâche sur une entité que de sélection d'une entité. Dans la procédure temps-réel, nous utiliserons au besoin, en plus de la fonction FTR , du théorème 4.3.1.2 et de la proposition 4.3.5.1, la définition de tâches urgentes introduite dans 3.3.1.4.

Sur la base de tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{21}).

Algorithme 4.3.5.2. Procédure en temps-réel OL-MTSA-FTP

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques et les premiers délais de maintenance préventive sur chaque entité du système comme définis dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-après.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer la procédure décrite dans 4.3.2.1 (AlgoOF-FTR ou AlgoOF-FTRW) à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche et le délai sur l'entité \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^2 à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

La proposition suivante donne la complexité de cette procédure de décision en temps-réel pour le processus de maintenance dans le cadre du problème (\mathcal{P}_{21}).

Proposition 4.3.5.3. Complexité

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-FTP est de l'ordre de $N' \left(\sum_{k=1}^N N_k\right)^2$.

La justification de cette proposition est analogue à celle de la complexité de la procédure OL-MTSA-W dans le chapitre 3.

Quelques tests numériques et analyses sur cette procédure en temps-réel sont l'objet de la sous-section suivante, avant d'aborder la configuration (\mathcal{P}_{22}) du problème (\mathcal{P}_2) qui prend en compte l'existence de temps logistique non négligeables entre les sites. Les conditions d'expérimentation des procédures en temps-réel pour la maintenance dans ce chapitre sont décrites ci-après.

4.3.6 Protocole d'expérimentation

Cette sous-section présente les conditions dans lesquelles les procédures décrites dans ce chapitre ont été expérimentées. Les tests numériques sur les approches présentées dans cette section et dans celles qui vont suivre ont été effectués sur des données aléatoirement générées. Les données présentées ici vont être utilisées dans l'ensemble des sections de ce chapitre.

Les données représentant les taux de défaillance (λ_{ik}) et de réparation (μ_{ik}) des entités sont celles utilisées dans le chapitre 3. Sur la base des taux de réparation, le temps moyen d'une tâche de maintenance sur chaque entité \mathcal{E}_{ik} (MTP_{ik}) dans le système est évalué comme étant son MTTR. Les premiers seuils critiques de disponibilité (α_{ik}^1) sont utilisés dans les mêmes conditions que dans le chapitre 3. Les durées τ_{ik} sont alors évaluées sur la base de la fonction disponibilité et de ces seuils de disponibilité. Au lieu du second seuil (α_{ik}^2) relatif aux délais sur les tâches de maintenance, nous avons généré les durées τ_{ik}^2 qui sont les durées depuis la complétion d'une tâche de maintenance sur l'entité \mathcal{E}_{ik} (supposées être confondues aux dates de remise en service de cette entité), jusqu'au délai sur la tâche de maintenance suivante sur cette même entité. Pour obtenir ces durées τ_{ik}^2 , nous avons généré des variables aléatoires ε_{ik} uniformes entre [0.25, 0.5]. Ensuite, la durée τ_{ik}^2 est exprimée comme $\tau_{ik}^2 = \tau_{ik} + MTP_{ik} + \varepsilon_{ik} \cdot MTP_{ik}$. La différence entre ces délais et les dates d'occurrence des tâches est représentée sur la figure 4.1(c).

Pour la configuration du système distribué (qui sera explicitement considéré comme tel dans la suite de ce chapitre), nous reconsidérons la configuration décrite dans le chapitre 3. Nous considérons donc un ensemble de sites sur lesquels la répartition des entités par site est uniforme discrète entre 20 et 80 entités par site (avec une moyenne de 50 entités par site). Par ailleurs, les temps logistiques entre les différents sites du système, temps logistiques qui seront explicitement pris en compte dans la suite, et que nous reconsidérerons explicitement dans la suite, sont pris suivant des valeurs absolues de la loi normale. Les figures 4.1 résument les distributions des données réels obtenues dans les générations et telles qu'utilisées dans ce chapitre.

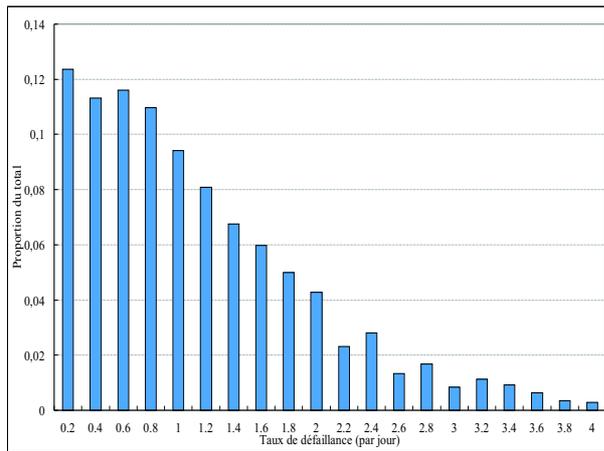
Comme dans le chapitre 3, les programmes sont toujours écrits en langage C et testés sur une station de travail Compaq AlphaServer ES40 DEC6600 opérant sous le système UNIX avec 2048 MB de mémoire RAM. Les tests sont tous entrepris sur un horizon \mathcal{H} de 365 jours (une année). Dans cette section comme dans les suivantes, les approches développées sont toujours programmées en deux versions. La première version prend en compte la règle d'optimalité locale en addition avec le concept d'urgence des tâches de maintenance. Dans une deuxième version, seule la règle d'optimalité locale est utilisée pour la prise de décision en temps-réel dans le processus de maintenance. De cette façon, les paramètres représentant les résultats sont toujours suivis d'un suffixe: U si le concept d'urgence est utilisé dans la prise de décision et NU dans le cas contraire.

Les résultats des expérimentations pour le problème (\mathcal{P}_{21}) (sans les temps logistiques), sur la base de ces données, sont représentés et analysés dans la sous-section suivante.

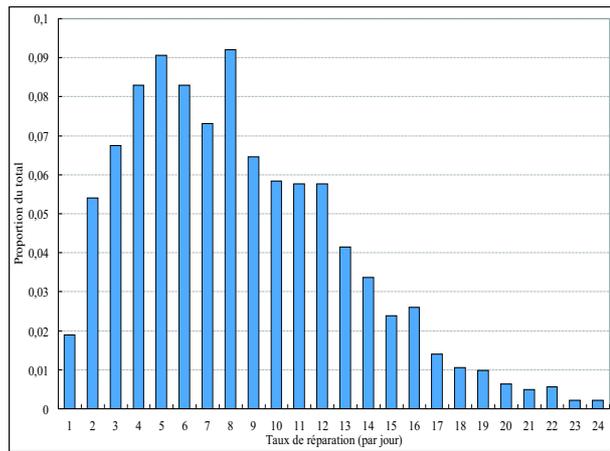
4.3.7 Résultats numériques

Expérimentations

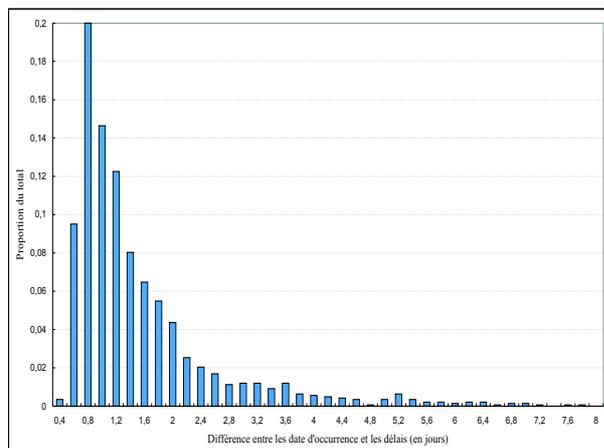
Les résultats présentés portent sur un système comprenant 500 entités au total. Les temps logistiques entre les sites étant faibles à ce niveau, ces entités sont considérées toutes ensemble formant un bloc. Comme dans le chapitre précédent, nous avons fait varier le nombre de ressources de maintenance entre 2% et 20% du nombre total d'entités, et ce nombre de ressources varie donc entre 10



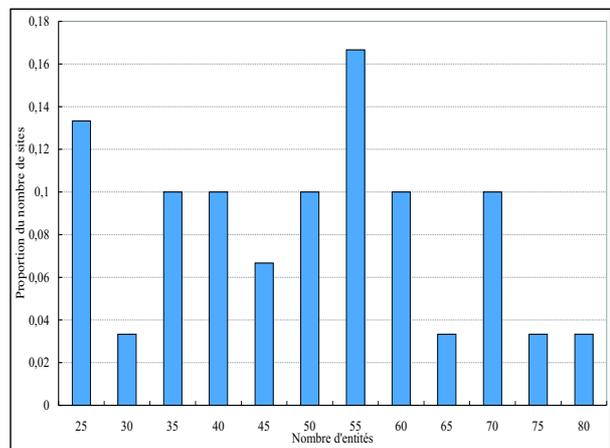
(a) Distribution des taux de défaillance



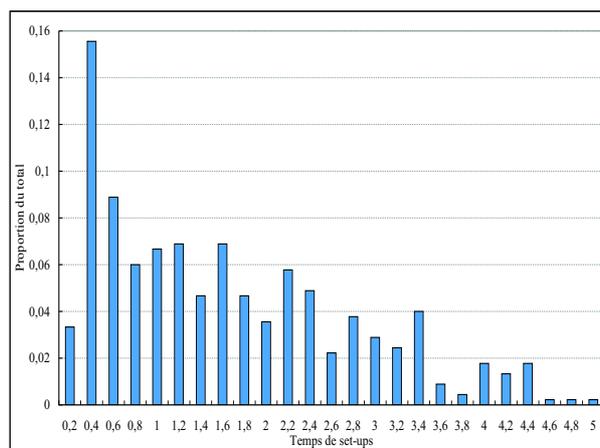
(b) Distribution des taux de réparation



(c) Distribution de la différence entre les dates d'occurrence et les délais



(d) Distribution du nombre d'entités par site



(e) Distribution des temps logistiques

Figure 4.1 – Distribution des données utilisées du système à maintenir avec les délais

et 100 ressources. Les résultats sont représentés en fonction de cette proportion du nombre d'entités que représente le nombre de ressources. La synthèse de ces résultats est présentée sur la figure 4.2.

Dans ces résultats, les paramètres DMI et DMS sur la figure 4.2(b), TMTpR sur la figure 4.2(e) et NMIpR sur la figure 4.2(f) ont les mêmes significations que dans le chapitre précédent. En revanche, le coût dépendant de deux aspects du système, les paramètres CMI-EcR sur la figure 4.2(a) représentent le coût moyen combiné des états critiques et des retards relevés sur les tâches de maintenance effectivement entreprises sur l'horizon \mathcal{H} des 365 jours alors que comme dans le chapitre précédent, CMS-EcR représente son correspondant sur l'ensemble du système. Ces coûts sont alors décomposés sur les figures 4.2(c) et 4.2(d). Sur la figure 4.2(c), le coût moyen CMS-EcR est décomposé en CMS-Ec (sa composante due uniquement aux états critiques) et CMS-R (sa composante due uniquement aux retards). De même sur la figure 4.2(d), le coût moyen CMI-EcR est décomposé en CMI-Ec (sa composante uniquement due aux états critiques) et CMI-R (sa composante due uniquement aux retards). Tous ces paramètres sont suivis de l'extension U lorsque le concept d'urgence est utilisé et NU dans le cas contraire.

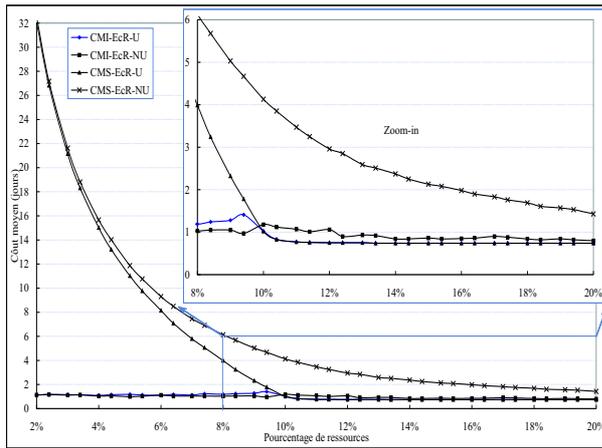
Analyse des résultats

Sur les figures 4.2(a), 4.2(b), 4.2(e) et 4.2(f), les mêmes observations se font que dans la section 3.3.4 du chapitre 3 à part l'échelle de coût moyen sur la figure 4.2(a) due au fait que ces coûts sont composés, en plus des états critiques, des retards des tâches de maintenance. Sur la figure 4.2(c), on voit que (avec l'utilisation du concept d'urgence des tâches) les coûts dus aux retards des tâches (aussi bien sa restriction aux tâches entreprises que sur le système tout entier) s'annulent en même temps que les délais d'interventions s'annulent à partir de 11% de ressources. À partir de ce moment, les coûts dus aux états critiques se stabilisent aussi, ce qui n'est pas le cas sans l'utilisation de l'urgence des tâches (voir sur les figures 4.2(c) et 4.2(d)). Pour cause de concision, nous inviterons le lecteur à consulter par exemple [4] pour davantage de détails sur cette approche.

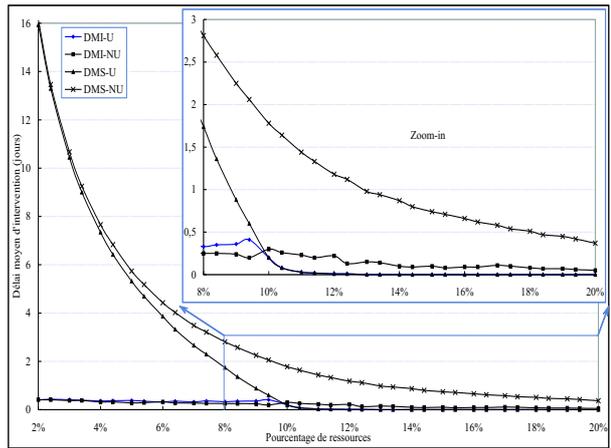
Nous passons maintenant à ce même problème avec des temps logistiques non négligeables entre les sites du système.

4.4 Coûts d'états critiques et retards avec des temps logistiques considérables

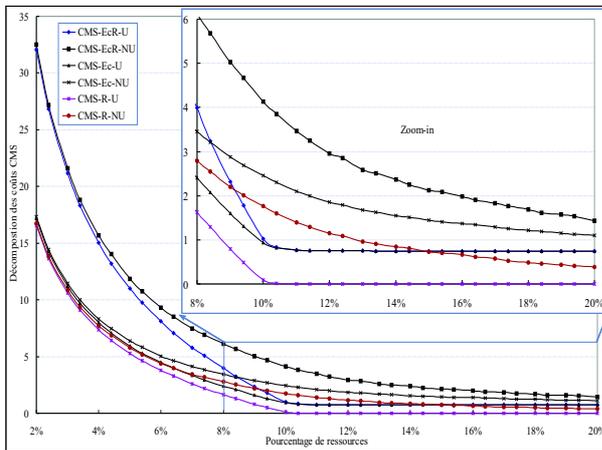
L'hypothèse de temps logistiques faibles, considérée dans le problème (\mathcal{P}_{21}) est maintenant relaxée dans cette section-ci. Nous considérons ici le problème (\mathcal{P}_2) dans sa configuration (\mathcal{P}_{22}). Les généralités du problème (\mathcal{P}_{22}) sont les mêmes que celles de la section précédente. Le critère de coût considéré est toujours une combinaison du coût des états critiques des entités et des retards des tâches de maintenance $C_2(\mathcal{H})$ (resp. $C_2^w(\mathcal{H})$) comme dans l'expression 4.2.7 (resp. 4.2.7). Ce coût est toujours considéré sur un horizon fini \mathcal{H} . En revanche, la configuration du système est telle que les temps logistiques entre les différents sites sont très importants et ne peuvent pas être sujets à des hypothèses réductrices.



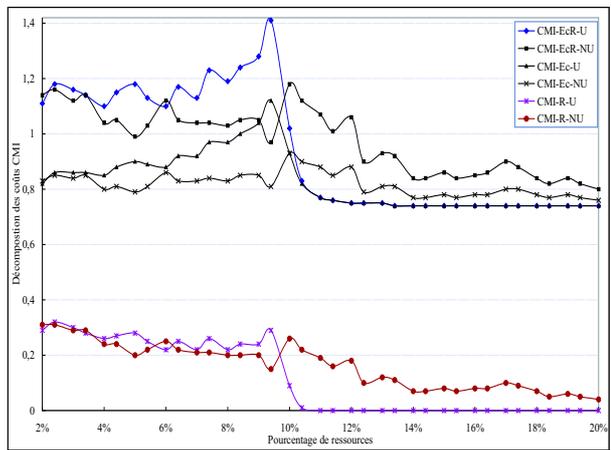
(a) Coût moyen de maintenance (en états critiques et retards)



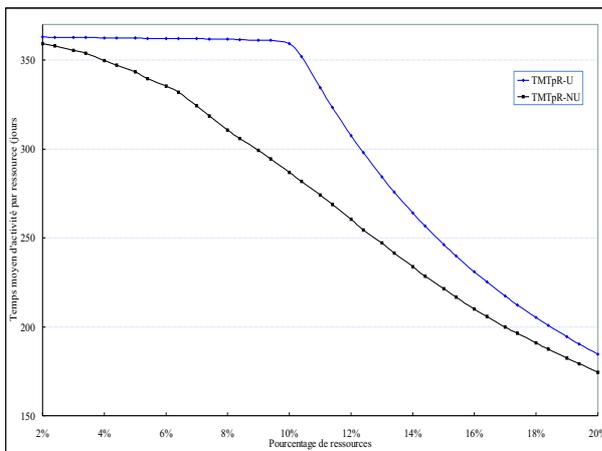
(b) Délai moyen d'intervention



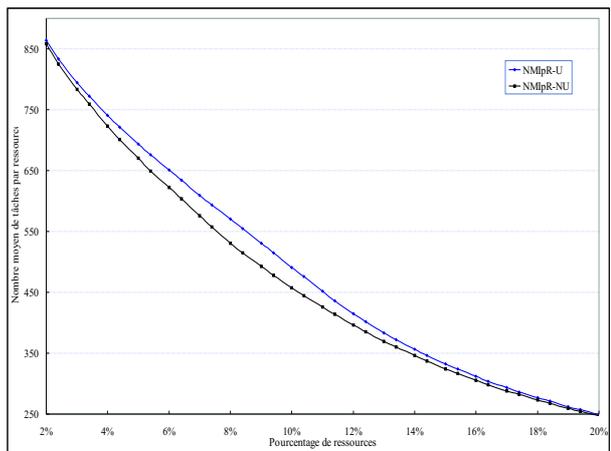
(c) Décomposition des coûts CMS



(d) Décomposition des coûts CMI



(e) Temps moyen d'activité par ressource



(f) Nombre moyen de tâches par ressource

Figure 4.2 – Résultats des coûts d'états critiques et de retards avec des temps logistiques faibles

Ainsi, le problème (\mathcal{P}_{22}) consiste à minimiser le coût combiné des états critiques des entités (correspondant au temps de réponse des tâches de maintenance) et des retards des tâches sur l'horizon \mathcal{H} avec des temps logistiques considérables entre les différents sites. Ces temps logistiques peuvent être symétriques ou non d'un site à un autre. Comme nous l'avons relevé plus haut, le problème (\mathcal{P}_{22}) est NP-difficile et c'est un problème de gestion temps-réel des tâches de maintenance. Pour permettre une décision adéquate à chaque instant, nous ferons des développements pour tirer des options de décisions localement optimales pour le critère de coût $C_2(\mathcal{H})$ (resp. $C_2^w(\mathcal{H})$) sur des problèmes localement statiques. Pour le problème statique, nous allons donc considérer un ensemble de n tâches (disponible ou à venir) dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . La sous-section suivante est dédiée au problème statique.

4.4.1 Le problème statique et les règles de décision

Considérons un ensemble de n tâches (disponibles ou à venir) dont on connaît les caractéristiques, qu'il faut ordonnancer sur une ressource. Chaque tâche i a une date d'occurrence r_i , une durée d'exécution p_i et un délai d_i . Un temps logistique s_{ij} est requis depuis la fin de la tâche i vers la tâche j suivante. Si une tâche i est exécutée en première position, alors elle est précédée d'un temps logistique s_{0i} où la tâche 0 est une tâche fictive dont la date de complétion est apparentée à la date initial de début de l'ordonnancement. Les temps logistiques s_{ij} dépendent de la séquence des tâches. Par ailleurs, à la fin d'une tâche i , le temps logistique s_{ij} vers une autre tâche j peut être exécuté avant même la date d'occurrence de cette dernière tâche. L'objectif est de minimiser la somme des temps de réponse des tâches et de leur retard par rapport à leur délais ($\sum_{i=1}^n ((c_i - r_i) + \max(0, c_i - d_i))$), ou de façon plus générale leur combinaison convexe ($\sum_{i=1}^n (W_f \cdot (c_i - r_i) + W_t \cdot \max(0, c_i - d_i))$). Pour cela, définissons la fonction suivante.

Définition 4.4.1.1. À tout instant t où une tâche k est complétée, la fonction *FTRS* (*Flow-time-Tardiness Rule with Setups*) est définie pour un couple de tâches (i, j) comme suit :

$$FTRS(i, j, k, t) = \max[PRTFST(i, j, k, t), Q_3(i, j, k, t)] + Q_4(i, k, t) + \max[Q_5(j), Q_6(i, j, k, t)] \quad (4.4.1)$$

avec

$$PRTFST(i, j, k, t) = 2\max(r_i, t + s_{ki}) + p_i + s_{ij},$$

$$Q_3(i, j, k, t) = \max(r_i, t + s_{ki}) + r_j,$$

$$Q_4(i, k, t) = \max[\max(r_i, t + s_{ki}), d_i - p_i],$$

$$Q_5(i) = \max(r_i, d_i - p_i)$$

$$Q_6(i, j, k, t) = \max(r_i, t + s_{ki}) + p_i + s_{ij}.$$

Avec cette fonction, le théorème suivant est vérifié.

Théorème 4.4.1.2. Soit π une séquence partielle de tâche terminée à un instant t par une tâche k . Pour tout couple de tâches (i, j) à ordonnancer à la date t , $i \preceq j$ si et seulement si

$$FTRS(i, j, k, t) \leq FTRS(j, i, k, t) \quad (4.4.2)$$

Preuve. La tâche i a les paramètres r_i , p_i , d_i et un temps logistique s_{ki} à partir de la fin de la tâche k . La tâche j a les paramètres r_j , p_j , d_j et un temps logistique s_{kj} à partir de la fin de la tâche k . Le

temps logistique de la tâche i vers la tâche j est s_{ij} est celui de j vers i est s_{ji} . Ces temps logistiques peuvent être symétriques ou pas.

En ordonnant les tâches i et j à l'instant t à la fin de la séquence partielle π , notons C_{ij}^k le coût additionnel engendré en ordonnant i directement avant j et C_{ji}^k celui en ordonnant directement j avant i . Alors, si on pose $R_i^k = \max(r_i, t + s_{ki})$ et $R_j^k = \max(r_j, t + s_{kj})$, nous avons

$$C_{ij}^k = \{[R_i^k + p_i - r_i] + [\max(R_i^k + p_i - d_i, 0)]\} + \{[\max(R_i^k + p_i + s_{ij}, r_j) + p_j - r_j] + [\max(\max(R_i^k + p_i + s_{ij}, r_j) + p_j - d_j, 0)]\}$$

et

$$C_{ji}^k = \{[R_j^k + p_j - r_j] + [\max(R_j^k + p_j - d_j, 0)]\} + \{[\max(R_j^k + p_j + s_{ji}, r_i) + p_i - r_i] + [\max(\max(R_j^k + p_j + s_{ji}, r_i) + p_i - d_i, 0)]\}.$$

Et ainsi, on a

$$\begin{aligned} C_{ij}^k - C_{ji}^k &= \{[R_i^k + p_i - r_i] + [\max(R_i^k + p_i - d_i, 0)]\} + \{[\max(R_i^k + p_i + s_{ij}, r_j) + p_j - r_j] + \\ &\quad [\max(\max(R_i^k + p_i + s_{ij}, r_j) + p_j - d_j, 0)]\} - \{[R_j^k + p_j - r_j] + [\max(R_j^k + p_j - d_j, 0)]\} \\ &\quad - \{[\max(R_j^k + p_j + s_{ji}, r_i) + p_i - r_i] + [\max(\max(R_j^k + p_j + s_{ji}, r_i) + p_i - d_i, 0)]\} \\ &= \{[R_i^k + p_i - r_i] + [\max(R_i^k + p_i + s_{ij}, r_j) + p_j - r_j]\} \\ &\quad + \{[\max(R_i^k + p_i - d_i, 0)] + [\max(\max(R_i^k + p_i + s_{ij}, r_j) + p_j - d_j, 0)]\} \\ &\quad - \{[R_j^k + p_j - r_j] + [\max(R_j^k + p_j + s_{ji}, r_i) + p_i - r_i]\} \\ &\quad - \{[\max(R_j^k + p_j - d_j, 0)] + [\max(\max(R_j^k + p_j + s_{ji}, r_i) + p_i - d_i, 0)]\} \\ &= F_{ij}^k + T_{ij}^k - F_{ji}^k - T_{ji}^k \end{aligned}$$

(4.4.3)

où

$$\begin{aligned} F_{ij}^k &= \{[R_i^k + p_i - r_i] + [\max(R_i^k + p_i + s_{ij}, r_j) + p_j - r_j]\} \\ F_{ji}^k &= \{[R_j^k + p_j - r_j] + [\max(R_j^k + p_j + s_{ji}, r_i) + p_i - r_i]\} \\ T_{ij}^k &= \{[\max(R_i^k + p_i - d_i, 0)] + [\max(\max(R_i^k + p_i + s_{ij}, r_j) + p_j - d_j, 0)]\} \\ T_{ji}^k &= \{[\max(R_j^k + p_j - d_j, 0)] + [\max(\max(R_j^k + p_j + s_{ji}, r_i) + p_i - d_i, 0)]\}. \end{aligned}$$

Or, d'une part

$$\begin{aligned} F_{ij}^k &= [R_i^k + p_i - r_i] + [\max(R_i^k + p_i + s_{ij}, r_j) + p_j - r_j] \\ &= R_i^k + \max(R_i^k + p_i + s_{ij}, r_j) + (p_i + p_j) - (r_i + r_j) \\ &= \max(R_i^k + R_i^k + p_i + s_{ij}, R_i^k + r_j) + (p_i + p_j) - (r_i + r_j) \\ &= \max(2R_i^k + p_i + s_{ij}, R_i^k + r_j) + (p_i + p_j) - (r_i + r_j). \end{aligned}$$

et, d'autre part

$$\begin{aligned} T_{ij}^k &= [\max(R_i^k + p_i - d_i, 0)] + [\max(\max(R_i^k + p_i + s_{ij}, r_j) + p_j - d_j, 0)] \\ &= [\max(R_i^k, d_i - p_i) + p_i - d_i] + [\max(\max(R_i^k + p_i + s_{ij}, r_j), d_j - p_i) + p_j - d_j] \\ &= \max(R_i^k, d_i - p_i) + \max(\max(R_i^k + p_i + s_{ij}, r_j), d_j - p_i) + (p_i + p_j) - (d_i + d_j) \\ &= \max(R_i^k, d_i - p_i) + \max(R_i^k + p_i + s_{ij}, \max(r_j, d_j - p_i)) + (p_i + p_j) - (d_i + d_j). \end{aligned}$$

Donc

$$F_{ij}^k - F_{ji}^k = \max(2R_i^k + p_i + s_{ij}, R_i^k + r_j) - \max(2R_j^k + p_j + s_{ji}, R_j^k + r_j)$$

et

$$T_{ij}^k - T_{ji}^k = [\max(R_i^k, d_i - p_i) + \max(R_i^k + p_i + s_{ij}, \max(r_j, d_j - p_i))] - [\max(R_j^k, d_j - p_j) + \max(R_j^k + p_j + s_{ji}, \max(r_i, d_i - p_i))].$$

Ainsi, l'égalité 4.4.3 devient

$$\begin{aligned} C_{ij}^k - C_{ji}^k &= [\max(2R_i^k + p_i + s_{ij}, R_i^k + r_j) - \max(2R_j^k + p_j + s_{ji}, R_j^k + r_j)] + [\max(R_i^k, d_i - p_i) + \max(R_i^k + p_i + s_{ij}, \max(r_j, d_j - p_i))] \\ &\quad - [\max(R_j^k, d_j - p_j) + \max(R_j^k + p_j + s_{ji}, \max(r_i, d_i - p_i))] \\ &= [\max(2R_i^k + p_i + s_{ij}, R_i^k + r_j) + \max(R_i^k, d_i - p_i) + \max(R_i^k + p_i + s_{ij}, \max(r_j, d_j - p_i))] - \\ &\quad [\max(2R_j^k + p_j + s_{ji}, R_j^k + r_j) + \max(R_j^k, d_j - p_j) + \max(R_j^k + p_j + s_{ji}, \max(r_i, d_i - p_i))] \end{aligned} \quad (4.4.4)$$

En utilisant la définition 4.4.1.1 de la fonction $FTRS$, l'égalité 4.4.4 est alors la même que

$$C_{ij}^k - C_{ji}^k = FTRS(i, j, k, t) - FTRS(j, i, k, t) \quad (4.4.5)$$

Ainsi, l'on a $i \preceq j$ si et seulement si $FTRS(i, j, k, t) \leq FTRS(j, i, k, t)$ \square

Remarque 4.4.1.3. *Si le coût est exprimé comme une somme pondérée du temps de réponse et du retard, avec le temps de réponse pondéré par W_f et le retard pondéré par W_t , alors le théorème 4.4.1.2 est toujours valable en remplaçant la fonction $FTRS$ par celle qui suit*

$$FTRSW(i, j, k, t) = W_f \cdot \max[PRTFST(i, j, k, t), Q_3(i, j, k, t)] + W_t \cdot Q_4(i, k, t) + W_t \cdot \max[Q_5(j), Q_6(i, j, k, t)] \quad (4.4.6)$$

Ainsi, tous les développements ultérieurs sont encore valables.

La relation de dominance définie par la fonction $FTRS$ (resp. $FTRSW$) n'est pas une relation d'ordre total dans l'ensemble des tâches à une date de décision. L'utilisation des fonctions $FTRS$ et $FTRSW$ se fera sur la base de la matrice de dominance et du vecteur des forces des tâches. L'algorithme de base de leur utilisation est l'objet de la prochaine sous-section.

4.4.2 Algorithme de base

Algorithme 4.4.2.1. Principe de base du $FTRS$ et du $FTRWS$: AlgoOF-FTRS, AlgoOF-FTRSW
Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée après la complétion d'une tâche k , les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$.
2. Calculer $FTRS(i, j, k, t)$ (ou $FTRSW(i, j, k, t)$ selon le cas) pour tous les couples de tâches (i, j) dans S .
3. Calculer la matrice de dominance Ω dans S .
4. Calculer le vecteur des forces (f_i) dans S .

5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
 Tant que $\text{card}(S) > 1$, aller à l'étape 4.

On notera que dans cette procédure, à chaque date de décision, la matrice des valeurs de la fonction FTRS (resp. FTRSW) et la matrice de dominance sont calculées une seule fois. Quand on réduit l'ensemble de recherche de la tâche à ordonnancer, les restrictions de ces matrices à ce nouvel ensemble sont considérées. Seul le vecteur des forces des tâches change lorsqu'on change d'ensemble de recherche.

Remarque 4.4.2.2. *Cet algorithme 4.4.2.1 peut aussi, comme les autres, être adapté dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égale à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant la plus petite indice parmi elles. Le choix des tâches se fait suivant les mêmes étapes que précédemment.*

La complexité de l'algorithme AlgoOF-FTRS (resp. AlgoOF-FTRSW) est donnée dans la proposition suivante. La démonstration de cette proposition est analogue est celle de l'Algorithme AlgoOF-PRTWF dans la proposition 3.4.3.3 du chapitre 3.

Proposition 4.4.2.3. *L'algorithme AlgoOF-FTRS (resp. AlgoOF-FTRWS) basé sur le principe de la fonction FTRS (resp. FTRSW), de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$ si n est le nombre de tâches à ordonnancer.*

Dans cette configuration, la question d'une borne inférieure est une question ouverte. Avant de passer à la procédure temps-réel pour le problème (\mathcal{P}_{22}) , nous illustrons dans la sous-section suivante le principe d'utilisation des fonctions FTRS et FTRSW.

4.4.3 Exemple 4.2 : illustration de la procédure AlgoOF-FTRS

Nous n'illustrons que la procédure avec la fonction FTRS, celle de la fonction FTRSW revenant à la même chose. Pour cet exemple 4.2, nous reprenons l'instance de l'exemple 4.1 dans le tableau 4.1. Nous complétons cette instance par les temps logistiques. Nous considérons pour cela le tableau 3.21 de temps logistiques non symétriques dans l'exemple 3.5 du chapitre 3. L'instance de l'exemple 4.2 est donc représentée sur les deux tableaux 4.6 et 4.7.

i	1	2	3	4	5
Date d'occurrence (r_i)	1,14	2,54	3,28	1,18	0,28
Durée d'exécution (p_i)	0,76	2,94	0,22	0,51	2,03
Délai (d_i)	1,98	5,78	3,52	1,74	2,51

Table 4.6 – Instance de l'exemple 4.2 pour les fonctions FTRS et FTRSW

À partir de ces données, les tableaux permettant la prise de décision à l'instant $t = 0$ vont être représentés. À l'instant $t = 0$, la tâche terminée est la tâche fictive 0. La matrice des valeurs de

$\downarrow i, j \rightarrow$	1	2	3	4	5
0	0,38	0,70	0,20	2,97	0,71
1	-	0,46	0,27	1,69	0,48
2	0,28	-	0,32	0,81	0,86
3	0,71	1,53	-	0,43	0,19
4	0,53	0,29	1,71	-	1,59
5	0,28	0,04	0,09	0,26	-

Table 4.7 – Matrice des temps logistiques s_{ij} dans l'exemple 4.2 avec la fonction FTRS

la fonction FTRS, la matrice de dominance et le vecteur des forces des tâches à cette date sont respectivement représentés dans les tableaux 4.8, 4.9 et 4.10.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	7,74	8,94	9,54	7,12
2	16,90	-	16,98	17,96	18,06
3	15,00	16,64	-	14,44	13,96
4	13,96	13,48	16,32	-	16,08
5	7,46	7,04	8,00	7,42	-

Table 4.8 – Valeurs de la fonction FTRS à $t = 0$ dans l'exemple 4.2

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	1	1
2	0	-	0	0	0
3	0	1	-	1	0
4	0	1	0	-	0
5	0	1	1	1	-

Table 4.9 – Matrice de dominance à $t = 0$ dans l'exemple 4.2

Tâche i	1	2	3	4	5
Force f_i	4	0	2	1	3

Table 4.10 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.2

La tâche sélectionnée à la date $t = 0$ sur la base du vecteur des forces des tâches est alors la tâche 1. Les autres décisions de l'exemple 4.2 sont représentées dans le tableau 4.11 et la séquence d'exécution des tâches est dans le tableau 4.12. Le coût moyen combiné de temps de réponse et de retard des tâches est alors de 5,458.

Après ces considérations sur le problème localement statique, nous reconsidérons maintenant le problème (\mathcal{P}_{22}) dans sa forme temps-réel sur la base de tout ce qui précède.

Instant de décision (t)	1,90	4,41	4,72	5,66
Tâche complétée	1	5	3	4
Ensemble des tâches	{2, 3, 4, 5}	{2, 3, 4}	{2, 4}	{2}
Tâches sélectionnée	5	3	4	2
Date de début (après set-up)	2,38	4,50	5,15	5,95

Table 4.11 – *Autres décisions dans l'exemple 4.2 pour la fonction FTRS*

Ordre d'exécution	1	2	3	4	5
Tâche	1	5	3	4	2
Date de début	1,14	2,38	4,50	5,15	5,95
Date de fin	1,90	4,41	4,72	5,66	8,89

Table 4.12 – *Séquence d'exécution des tâches de l'exemple 4.2*

4.4.4 Algorithme temps-réel pour la résolution du problème (\mathcal{P}_{22})

En plus de tout ce qui précède, nous allons utiliser la proposition suivante dans la procédure en temps-réel. Cette proposition nous permettra par la suite, tout comme dans les précédents problèmes, de parler indifféremment des entités que des demandes de tâches de maintenance sur ces entités.

Proposition 4.4.4.1. *En respectant le théorème 4.4.1.2 avec la fonction FTRS (et aussi avec la fonction FTRSW), on respecte aussi l'ordre d'arrivée des demandes de tâches de maintenance sur une entité quelconque \mathcal{E}_{ik} . En d'autres termes, pour deux entiers m_1 et m_2 tels que $m_1 < m_2$, le théorème 4.4.1.2 permet de programmer la $m_1^{\text{ième}}$ tâches sur \mathcal{E}_{ik} avant la $m_2^{\text{ième}}$.*

La justification de cette proposition est analogue à celle de la proposition 3.3.1.3 dans le chapitre 3. Il suffira pour cela de considérer une entité quelconque \mathcal{E}_{ik} , deux entiers m_1 et m_2 tels que $m_1 < m_2$ (avec deux tâches sur \mathcal{E}_{ik} supposées avoir la même durée d'exécution) et un temps logistique nul entre les deux tâches, de considérer que les dates d'occurrence des tâches et les délais sont tels que $r_{m_1,ik} < r_{m_2,ik}$, $d_{m_1,ik} < d_{m_2,ik}$ et de comparer les valeurs de la fonction FTRS.

Ainsi, dans la suite de cette section, nous parlerons indifféremment de sélection d'une tâche sur une entité que de sélection d'une entité. Dans la procédure temps-réel, nous utiliserons, en plus de la fonction FTRS, du théorème 4.4.1.2 et de la proposition 4.3.5.1, la définition 3.3.1.4 d'urgence de tâches de maintenance.

En se basant sur tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{22}).

Algorithme 4.4.4.2. Procédures temps-réel OL-MTSA-FTPS

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques et les premiers délais de maintenance préventive sur chaque entité du système comme définis dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer la procédure décrite dans 4.4.2.1 (AlgoOF-FTRS et AlgoOF-FTRSW) à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche et le délai sur l'entité \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^2 à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

Proposition 4.4.4.3. *Complexité*

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-FTPS est de l'ordre de $N' \left(\sum_{k=1}^N N_k\right)^2$.

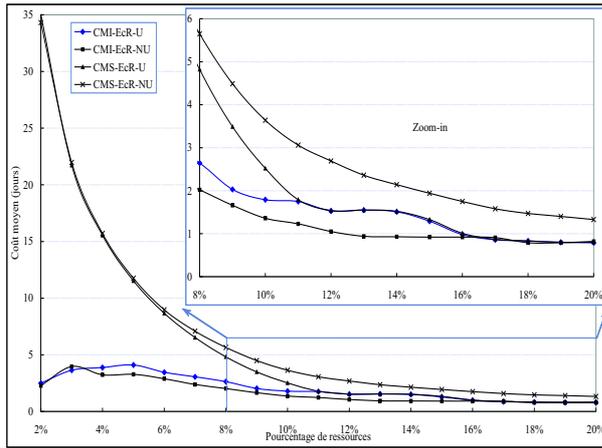
La justification de cette proposition est analogue à celle de la complexité de la procédure OL-MTSA-W dans le chapitre 3.

La sous-section suivant montre quelques résultats d'expérimentations et analyse sur le problème (\mathcal{P}_2) avec des temps logistiques importants entre les différents sites, sur la base de tous les développements ci-dessus.

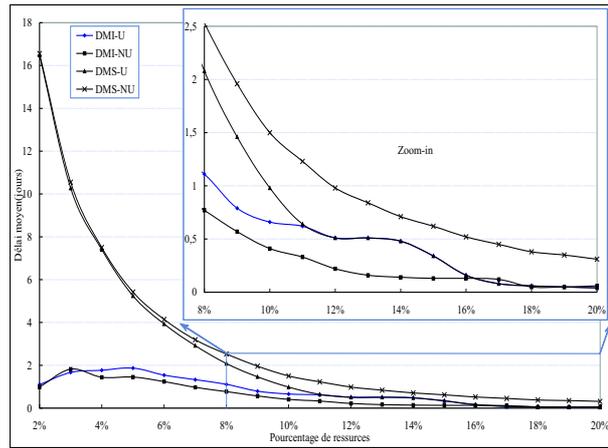
4.4.5 Résultats numériques

Expérimentations

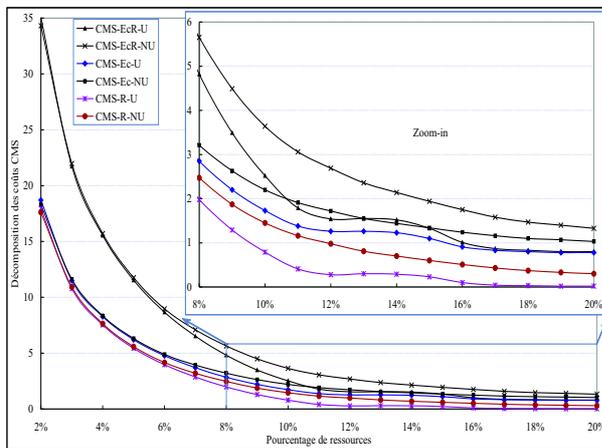
Pour les expérimentations, nous reconsidérons le protocole d'expérimentation décrit dans la sous-section 4.3.6 avec à présent les données relatives aux temps logistiques entre les sites. Les résultats d'expérimentation montrés ici sont relatifs, comme dans le chapitre précédent, à un ensemble de 10 sites. Le nombre moyen d'entités par site étant de 50, le nombre moyen d'entités dans le système composé des 10 sites est de 500 entités. Mais le nombre réel d'entités par site est aléatoire uniforme entre 20 et 80, et le nombre réel d'entités dénombrées dans le système des 10 sites est de 441 entités. Les résultats sont représentés sur la figure 4.3. Sur les figures 4.3(a), 4.3(b), 4.3(c) 4.3(d) et 4.3(f), les paramètres ont les mêmes significations que dans la section précédente alors que sur la figure 4.3(e), ils ont la signification des temps d'utilisation par ressource dans les cas de temps logistiques considérables dans le chapitre précédent.



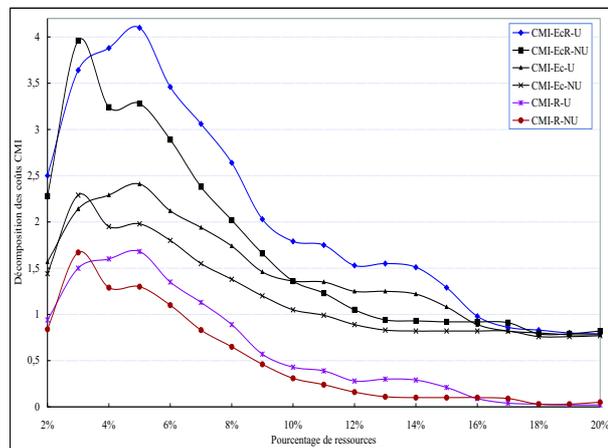
(a) Coût moyen de maintenance (en états critiques et retards)



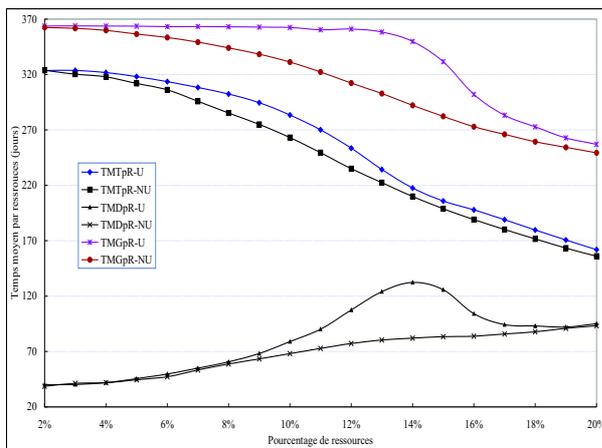
(b) Délai moyen d'intervention



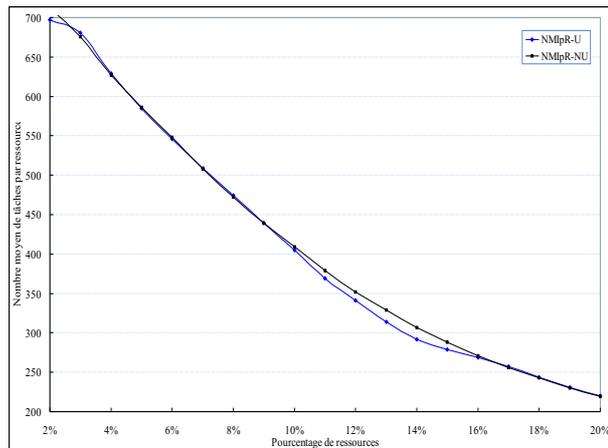
(c) Décomposition des coûts CMS



(d) Décomposition des coûts CMI



(e) Temps moyen d'activité par ressource



(f) Nombre moyen de tâches par ressource

Figure 4.3 – Résultats des coûts d'états critiques et de retards avec des temps logistiques importants

Analyse des résultats

Pour les figures 4.3(a), 4.3(b), 4.3(c) et 4.3(d), les observations sont analogues à celles la section précédente à part le fait : que la convergence des coûts sur les tâches de maintenance effectivement entreprises sur l'horizon et ceux sur le système tout entier (lorsque le concept d'urgence est utilisé) se réalise à 11% de ressources au lieu de 10% pour la configuration précédente ; que les délais moyens ne s'annulent pas après la convergence des deux types de coûts. Pour les figures 4.3(e) et 4.3(f), les observations sont analogues à celles de la sous-section 3.5.6 du chapitre 3. Pour davantage de détails, le lecteur peut se référer à [6] par exemple.

Nous abordons maintenant le second type de problème qui met en jeu des contraintes de délai sur les tâches de maintenance. Ce problème qui implique des pénalités d'avance et de retard des tâches de maintenance est le problème (\mathcal{P}_3). Nous l'aborderons dans les deux sections qui suivent. Dans la section ci-après, le problème (\mathcal{P}_3) est considéré dans sa configuration (\mathcal{P}_{31}) avec des temps logistiques faibles entre les différents sites du système.

4.5 Coûts d'avances et retards avec des temps logistiques faibles

Considérons maintenant le problème (\mathcal{P}_3) qui implique des coûts des avances et des retards des tâches de maintenance comme expliqué dans la position du problème dans la sous-section 4.2.2. Le coût $\mathcal{C}_3(\mathcal{H})$ (resp. $\mathcal{C}_3^w(\mathcal{H})$) dans l'expression 4.2.9 (resp. 4.2.11) est dû aux avances et aux retards des tâches de maintenance par rapport à des dates requises et doit être minimisé sur l'horizon fini \mathcal{H} . Dans cette section, nous intéressons au problème (\mathcal{P}_3) dans sa configuration (\mathcal{P}_{31}) avec des temps logistiques faibles (et donc assimilés à 0) entre les différents sites.

Comme révélé plus haut, le problème (\mathcal{P}_{31}) est, comme les problèmes (\mathcal{P}_{21}) et (\mathcal{P}_{22}), NP-difficile et c'est un problème d'organisation en temps-réel des tâches de maintenance. Nous commencerons donc par des développements localement optimaux sur des problèmes localement statiques afin de trouver des outils pour la prise de décision à tout instant sur l'horizon \mathcal{H} dans l'optique de minimiser le coût $\mathcal{C}_3(\mathcal{H})$ ou $\mathcal{C}_3^w(\mathcal{H})$ de façon plus générale. Nous allons donc considérer un ensemble de n tâches dont on connaît les caractéristiques, à ordonnancer à un instant donné sur une ressource. La sous-section suivante est dédié au problème statique.

4.5.1 Le problème statique et les règles de décision

Soit donc un ensemble de n tâches (disponible ou à venir), dont on connaît les caractéristiques, à ordonnancer sur une ressource à un instant t . Chaque tâche i a une date d'occurrence r_i (qui diffère d'une tâche à une autre), une durée d'exécution p_i et un délai d_i . Les tâches sont exécutées sans préemption. Contrairement à ce qui se passe dans les problèmes (\mathcal{P}_{21}) et (\mathcal{P}_{22}), les délais ici ne sont pas simplement à ne pas dépasser. En fait, une tâche i terminée avant sa date d_i engendre un coût d'avance et, après d_i , elle engendre un coût de retard. L'objectif est donc de minimiser leur coût combiné des avances et des retards par rapport aux délais d_i ($\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i))$), ou de façon plus générale ($\sum_{i=1}^n (W_e \cdot \max(0, d_i - c_i) + W_r \cdot \max(0, c_i - d_i))$) qui est une combinaison

convexe de ces deux coûts, où c_i est la date de complétion de la tâche i . Dans les développements qui vont suivre, le coût $(\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i)))$ sera principalement considéré. Posons donc la définition suivante dans l'optique de la résolution du problème statique.

Définition 4.5.1.1. *À tout instant t à la fin d'une séquence partielle de tâches, la fonction ETR (Earliness-Tardiness Rule) est définie pour une paire de tâches $\{i, j\}$ comme suit :*

$$ETR(i, j, t) = |\max(r_i, t) + p_i - d_i| + |\max(\max(r_i, t) + p_i, r_j) + p_j - d_j| \quad (4.5.1)$$

On a alors le théorème suivant :

Théorème 4.5.1.2. *Soit un ensemble de tâches à ordonnancer sur une ressource afin de minimiser le coût des avances et retards $(\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i)))$ en présence de dates d'occurrence inégales des tâches. Soit une séquence partielle de tâche π terminée à une date t . Pour tout couple de tâches (i, j) et par rapport au coût ci-dessus, $i \preceq j$ si et seulement si*

$$ETR(i, j, t) \leq ETR(j, i, t) \quad (4.5.2)$$

Preuve. La tâche i a les paramètres r_i , p_i et d_i (qui sont respectivement sa date d'occurrence, sa durée d'exécution, et sa date d'échéance qui est la date d'achèvement pour un coût minimum). De même, la tâche j a les paramètres r_j , p_j et d_j . La date de décision est t .

En ordonnant les tâches i et j à l'instant t à la fin de la séquence partielle π , notons C_{ij}^3 le coût additionnel engendré en ordonnant i directement avant j , et C_{ji}^3 celui en ordonnant directement j avant i . Posons à la date t $R_i = \max(r_i, t)$ et $R_j = \max(r_j, t)$. Nous avons alors

$$C_{ij}^3 - C_{ji}^3 = (|R_i + p_i - d_i| + |\max(R_i + p_i, r_j) + p_j - d_j|) - (|R_j + p_j - d_j| + |\max(R_j + p_j, r_i) + p_i - d_i|)$$

Ainsi, $C_{ij}^3 \leq C_{ji}^3$ si et seulement si

$$(|R_i + p_i - d_i| + |\max(R_i + p_i, r_j) + p_j - d_j|) \leq (|R_j + p_j - d_j| + |\max(R_j + p_j, r_i) + p_i - d_i|)$$

Une simplification de cette inégalité aurait été préférable mais en son absence, nous l'utilisons sous cette forme. Avec la matrice de dominance, cela est possible. Ainsi, avec l'utilisation de la fonction ETR dans la définition 4.5.1.1, $i \preceq j$ si et seulement si $ETR(i, j, t) \leq ETR(j, i, t)$ \square

Il faut noter qu'avec l'usage de la matrice de dominance et du vecteur des forces des tâches introduits dans le chapitre 3, lorsqu'il est possible de déduire d'un coût une règle d'optimalité locale prenant en compte à chaque fois une seule tâche (comme l'est par exemple la fonction PRTF), c'est toujours préférable. Cependant, dans le cas où cela n'est pas possible, ces matrices de dominance et ces vecteurs de forces des tâches permettent l'usage de règles moins simples. C'est ce qui nous permet de traiter les différents cas abordés jusqu'ici.

Remarque 4.5.1.3. *Le coût pourrait être exprimé comme une somme pondérée des pénalités d'avance et du retard, avec l'avance pondérée par W_e et le retard pondéré par W_t . Dans ce cas, le théorème 4.5.1.2 est toujours valable en remplaçant la fonction ETR par la suivante*

$$ETRW(i, j, t) = W_e \cdot |\max(r_i, t) + p_i - d_i| + W_t \cdot |\max(\max(r_i, t) + p_i, r_j) + p_j - d_j| \quad (4.5.3)$$

Tous les développements ultérieurs sont alors toujours valables.

Remarque 4.5.1.4. Si le coût minimal pour la tâche i est atteint sur une tranche horaire $[e_i, d_i]$ plutôt qu'à une date précise d_i , on peut remplacer la règle ETR par $\max(e_i - (R_i + p_i), 0) + \max((R_i + p_i) - d_i, 0) + \max(e_j - (\max(R_i + p_i, r_j) + p_j), 0) + \max((\max(R_i + p_i, r_j) + p_j) - d_j, 0)$ où $R_i = \max(r_i, t)$ et $R_j = \max(r_j, t)$. Les procédures décrites ici peuvent alors être toujours utilisées.

Une borne inférieure du coût des avances et des retards en présence de dates d'occurrence inégales des tâches pourrait être obtenue en combinant la borne inférieure du retard en présence de date d'occurrence inégales utilisée dans le problème (\mathcal{P}_{21}) avec une borne inférieure possible du coût des avances en présence de dates d'occurrence inégales des tâches. Le principe de base de l'utilisation de la fonction ETR est donné dans ci-après.

4.5.2 Algorithme de base

Ce principe de base est d'abord fait pour une ressource.

Algorithme 4.5.2.1. Principe de base du ETR et du ETRW : AlgoOF-ETR, AlgoOF-ETRW
Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée, les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$
2. Calculer $ETR(i, j, t)$ (ou $ETRW(i, j, t)$ selon le cas) pour toutes les paires de tâches i, j dans S .
3. Calculer la matrice de dominance Ω dans S .
4. Calculer le vecteur des forces (f_i) dans S .
5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
 Tant que $\text{card}(S) > 1$, aller à l'étape 4.

Remarque 4.5.2.2. Cet algorithme 4.5.2.1 peut aussi, comme les autres, être adapté dans le cas de ressources parallèles, avec un nombre M de ressources supérieur ou égal à deux. Dans ce cas, l'instant t de décision est la date de disponibilité au plus tôt d'une ressource. S'il y a plusieurs ressources disponibles à cette date, la ressource sélectionnée est celle ayant le plus petit indice parmi elles. La sélection des tâches se fait suivant les mêmes étapes que précédemment.

La complexité de l'algorithme AlgoOF-ETR (resp. AlgoOF-ETRW) est donnée dans la proposition suivante. La démonstration de cette proposition est analogue est celle de l'Algorithme AlgoOF-PRTWF dans la proposition 3.4.3.3 du chapitre 3.

Proposition 4.5.2.3. L'algorithme AlgoOF-ETR (resp. AlgoOF-ETRW) basé sur le principe de la fonction ETR (resp. ETRW), de la matrice de dominance et du vecteur des forces a une complexité en $O(n^3)$ si n est le nombre de tâches à ordonnancer.

Pour ce problème statique, Valente et Alves [157] ont adapté des bornes inférieures (à partir de problèmes où les tâches ont les mêmes dates d'occurrence) en utilisant des bornes inférieures pour la pénalité d'avance et des bornes inférieures des pénalités de retard. Avant de passer à l'utilisation de ces règles pour le problème, nous allons les illustrer.

4.5.3 Exemple 4.3 illustrant la procédure AlgoOF-ETR

A cause de la ressemblance d'utilisation entre les fonctions ETR et ETRW, nous n'illustrons que la procédure avec la fonction ETR. Pour l'illustration de cette procédure avec la fonction ETR, nous considérons l'instance de l'exemple 4.1 avec la fonction FTR. Mais ici, le coût d'avances et de retards des tâches est considéré alors que dans l'exemple 4.1, le coût de temps de réponse et de retard était l'objet. L'instance de cet exemple 4.3 est donc la même que sur le tableau 4.1.

La matrice des valeurs de la fonction ETR, la matrice de dominance et le vecteur des forces des tâches à l'instant $t = 0$ sont représentés respectivement sur les tableaux 4.13, 4.14 et 4.15.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	0,38	0,10	0,75	1,50
2	4,56	-	2,48	4,55	5,30
3	2,30	0,68	-	2,29	3,04
4	0,52	0,35	0,07	-	1,26
5	1,29	0,50	0,22	1,28	-

Table 4.13 – Valeurs de la fonction ETR à $t = 0$ dans l'exemple 4.3

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	0	0
2	0	-	0	0	0
3	0	1	-	0	0
4	1	1	1	-	1
5	1	1	1	0	-

Table 4.14 – Matrice de dominance à $t = 0$ dans l'exemple 4.3

Tâche i	1	2	3	4	5
Force f_i	2	0	1	4	3

Table 4.15 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.3

La tâche 4 est sélectionnée à cette date. Les autres décisions se font de la même façon en ne considérant à chaque décision que les tâches non encore ordonnancées. La séquence d'exécution des tâches dans l'exemple 4.3 est alors représentée sur le tableau 4.16 avec un coût moyen combiné des avances et retards des tâches égal à 1,25.

Tous ces outils vont être maintenant utilisés dans la procédure temps-réel pour le problème (\mathcal{P}_{31}).

Ordre d'exécution	1	2	3	4	5
Tâche	4	1	3	5	2
Date de début	1,18	1,69	3,28	3,50	5,53
Date de fin	1,69	2,45	3,50	5,53	8,47

Table 4.16 – Séquence d'exécution des tâches de l'exemple 4.3 avec la fonction ETR

4.5.4 Algorithme en temps-réel pour la résolution du problème (\mathcal{P}_{31})

Nous reconsidérons maintenant le problème (\mathcal{P}_{31}) dans sa forme temps-réel. Tout comme dans les précédents problèmes, nous parlerons indifféremment dans la procédure en temps-réel de cette section, de sélection d'une tâche sur une entité ou de sélection de l'entité elle-même. Dans cette procédure temps-réel, nous utiliserons, en plus de la fonction ETR et du théorème 4.5.1.2, la définition 3.3.1.4 de tâches urgentes dans le chapitre 3.

Sur la base de tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{31}).

Algorithme 4.5.4.1. Procédure en temps-réel OL-MTSA-ETP

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques et les premiers délais de maintenance préventive sur chaque entité du système comme définis dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-après.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer la procédure décrite dans 4.5.2.1 (AlgoOF-ETP et AlgoOF-ETRW) à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche et la date suivante de maintenance de coût minimal de \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^3 à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

La proposition suivante donne la complexité de cette procédure temps-réel.

Proposition 4.5.4.2. Complexité

Si N' est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-FTP est de l'ordre de $N' \left(\sum_{k=1}^N N_k \right)^2$.

La justification de cette proposition est analogue à celle de la complexité de la procédure OL-MTSA-W dans le chapitre 3.

Des résultats d'expérimentation sont présentés ci-après sur cette configuration du problème (\mathcal{P}_{31}).

4.5.5 Résultats numériques

Expérimentations

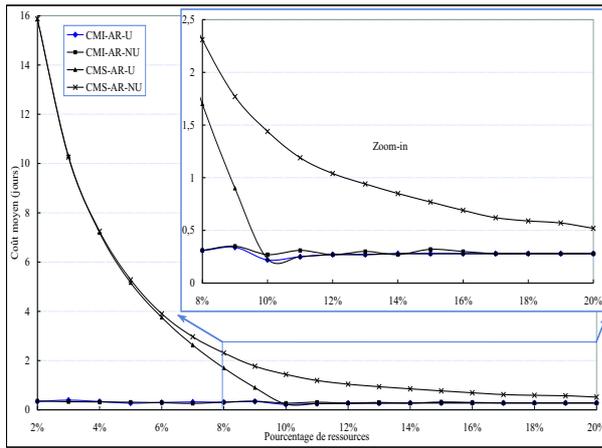
Les expérimentations sont conduites dans les mêmes conditions que dans la section 4.3. Les résultats portent sur un système comprenant 500 entités au total. Les temps logistiques ne sont pas considérés et les entités du système sont prises en bloc. Nous avons fait varier le nombre de ressources de maintenance entre 2% et 20% du nombre total d'entités et ce nombre de ressources varie donc entre 10 et 100 ressources. Les résultats sont représentés en fonction de cette proportion du nombre d'entités que représente le nombre de ressources. La synthèse de ces résultats est présentée sur la figure 4.4.

Les paramètres des figures 4.4(b), 4.4(e) et 4.4(f) ont les mêmes significations que dans la section 4.3. Par contre, comme ici le coût dépend des avances et des retards des tâches de maintenance, les paramètres CMI-AR sur la figure 4.4(a) représentent le coût moyen combiné des avances et des retards relevés sur les tâches de maintenance effectivement entreprises sur l'horizon \mathcal{H} des 365 jours alors que CMS-AR représente son correspondant sur l'ensemble du système. Ces deux coûts moyens sont décomposés sur les figures 4.4(c) et 4.4(d). Sur la figure 4.4(c), le coût moyen CMS-AR est décomposé en CMS-A (sa composante due uniquement aux avances) et CMS-R (sa composante due uniquement aux retards). De même sur la figure 4.4(d), le coût moyen CMI-AR est décomposé en CMI-A (sa composante uniquement due aux avances) et CMI-R (sa composante due uniquement aux retards). Comme dans les cas précédents, tous ses paramètres sont suivis de l'extension U lorsque le concept d'urgence est utilisé et NU dans le cas contraire.

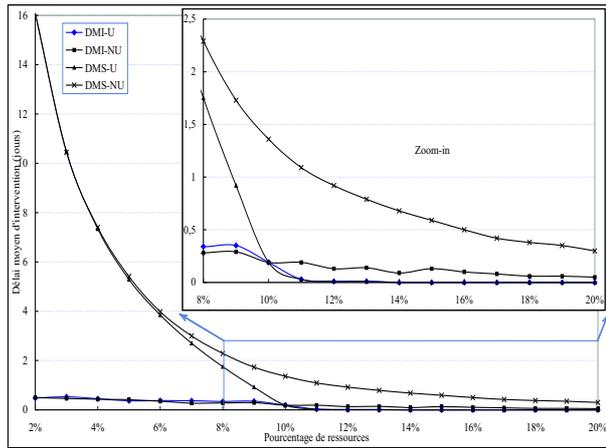
Analyse des résultats

Pour les figures 4.4(b), 4.4(e) et 4.4(f) nous allons toujours renvoyer à la sous-section 3.3.4 du chapitre 3.

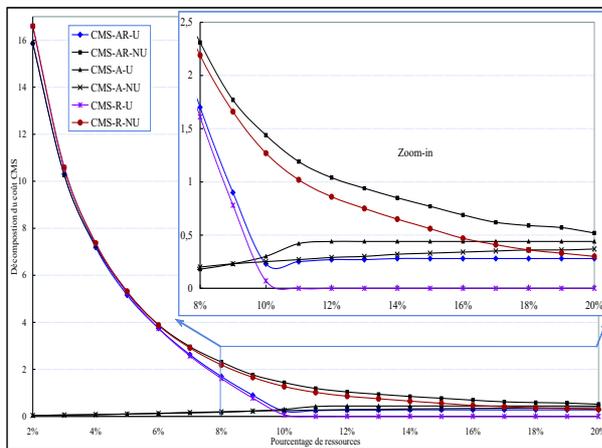
Sur les courbes 4.4(a) et 4.4(b) on peut faire diverses observations. La convergence des coûts moyens CMI et CMS (courbes 4.4(a)) en utilisant le concept d'urgence des tâches de maintenance, est réalisée comme c'était le cas dans les sections antérieures autour de 10% de ressources. Avant la convergence des coûts CMI et CMS (point 10% de ressources), la composante du coût moyen due aux retards (sur le système tout entier) est la composante principale du coût (voir 4.4(c)). La composante du coût due aux avances des tâches de maintenance étant très faible, la composante due aux retards est alors très proche du coût moyen global.



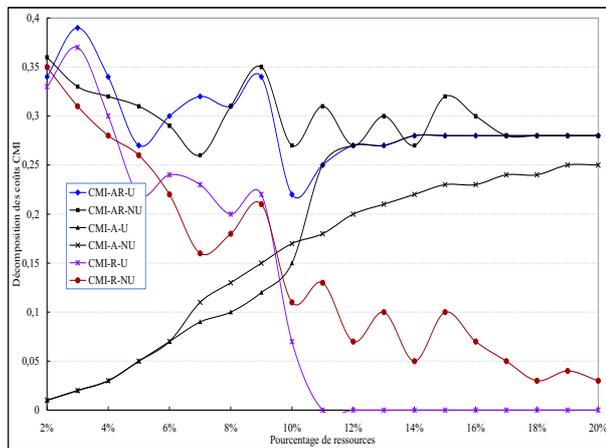
(a) Coût moyen d'avances et retards



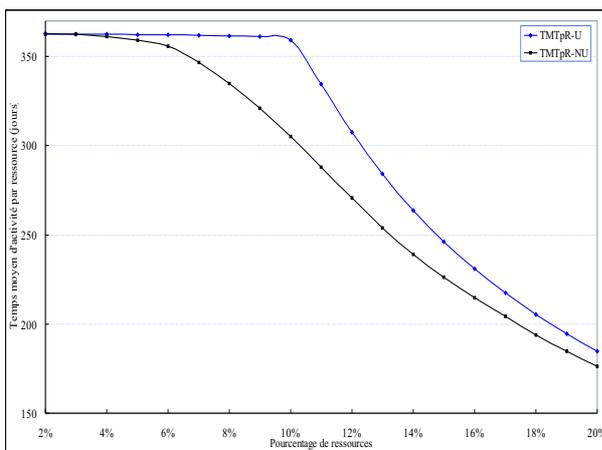
(b) Délai moyen d'intervention



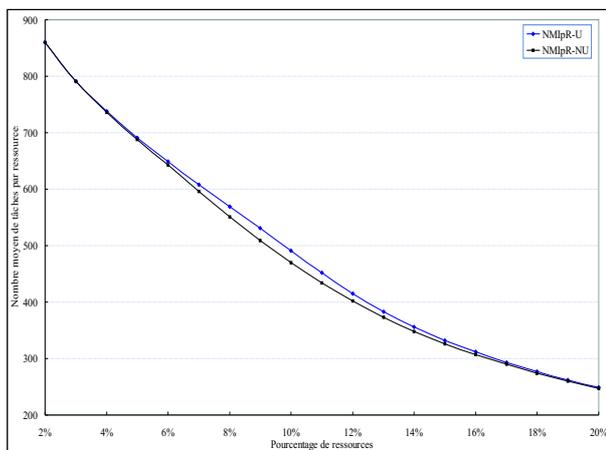
(c) Décomposition des coûts CMS



(d) Décomposition des coûts CMI



(e) Temps moyen d'activité par ressource



(f) Nombre moyen de tâches par ressource

Figure 4.4 – Résultats des coûts d'avance et de retards avec des temps logistiques faibles

Mais après le point de convergence (10% de ressources), le coût dû aux retards s'annule en même temps que les délais d'intervention s'annulent. Et donc à partir de ce point de convergence (symbolisant un nombre minimal de ressources nécessaires suivant la taille et la configuration du système), le système ne souffre plus (avec le concept d'urgence des tâches) de coûts dus aux retards des tâches de maintenance. On avait déjà observé ce fait dans le cas des coûts d'états critiques et des retards, avec des temps logistiques faibles.

Dans ce cas aussi, le coût engendré sur le système lorsque le concept d'urgence des tâches n'est pas utilisé est toujours bien supérieur au cas contraire. Mais on ne se serait pas douter de cela au début, vu que l'urgence des tâches est définie par rapport aux états critiques et non par rapport aux avances et retards des tâches de maintenance. Tout cela signifie que le concept d'urgence est un outil qu'il est nécessaire de combiner aux approches de prise de décision en temps-réel dans le processus de maintenance.

Nous passons maintenant à ce problème avec des temps logistiques non négligeables entre les sites.

4.6 Coûts des avances et retards avec des temps logistiques importants

Nous reconsidérons le problème (\mathcal{P}_3) sous sa configuration plus générale qu'est (\mathcal{P}_{32}). Nous n'allons pas revenir sur les généralités du problème (\mathcal{P}_{32}) qui sont les mêmes que celles du problème (\mathcal{P}_{31}). Le coût en jeu est toujours le coût $C_3(\mathcal{H})$ (resp. $C_3^w(\mathcal{H})$) des avances et des retards des tâches de maintenance par rapport à des dates données comme dans l'expression 4.2.9 (resp. 4.2.11) qui est à minimiser sur l'horizon fini \mathcal{H} . Mais dans le problème (\mathcal{P}_{31}), les temps logistiques entre les différents sites du système sont supposés faibles et assimilés à 0. Maintenant cette hypothèse est relaxée dans le problème (\mathcal{P}_{32}). Il existe donc un temps logistique important s_{kl} d'un site \mathcal{S}_k à un autre site \mathcal{S}_l . Ces temps logistiques peuvent être symétriques ou non d'un site à un autre. Le problème (\mathcal{P}_{32}), tout comme le problème (\mathcal{P}_{31}) est un problème NP-difficile de gestion temps-réel. Nous allons donc, dans un premier temps procéder à des développements pour des options de décision localement optimales sur des problèmes localement statiques. Ces développements sur le cas localement statiques sont l'objet de la sous-section suivante.

4.6.1 Le problème statique et les règles de décision

Soit un ensemble de n tâches (disponibles ou à venir) dont on connaît les caractéristiques, qu'il faut ordonnancer sur une ressource. Chaque tâche i a une date d'occurrence r_i , une durée d'exécution p_i et un délai d_i . Les tâches sont exécutées sans préemption. Comme dans le problème (\mathcal{P}_{31}), les délais ici ne sont pas simplement à ne pas dépasser. Une tâche i terminée avant sa date d_i engendre un coût d'avance et après d_i , elle engendre un coût de retard. Par ailleurs, un temps logistique s_{ij} est requis depuis la fin de la tâche i vers la tâche j suivante. Si une tâche i est exécutée en première position, alors elle est précédée d'un temps logistique s_{0i} où la tâche 0 est une tâche fictive dont la date de complétion est apparentée à la date initial de début du

processus. Les temps logistique s_{ij} dépendent de la séquence des tâches. À la fin d'une tâche i , le temps logistique s_{ij} vers une autre tâche j peut être exécuté avant même la date d'occurrence de cette dernière tâche. L'objectif est de minimiser leur coût combiné des avances et des retards par rapport aux délais d_i ($\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i))$) ou, de façon plus générale ($\sum_{i=1}^n (W_e \cdot \max(0, d_i - c_i) + W_t \cdot \max(0, c_i - d_i))$) qui est une combinaison convexe de ces deux coûts, où c_i est la date de complétion de la tâche i . Le coût ($\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i))$) sera principalement considéré. Posons la définition suivante qui nous aidera dans la résolution.

Définition 4.6.1.1. *À tout instant t à la complétion d'une séquence partielle de tâches terminée par une tâche k , la fonction ETRS (Earliness-Tardiness Rule with Set-up times) est définie pour une paire de tâches $\{i, j\}$ comme suit :*

$$ETRS(i, j, k, t) = |\max(r_i, t + s_{ki}) + p_i - d_i| + |\max(\max(r_i, t + s_{ki}) + p_i + s_{ij}, r_j) + p_j - d_j| \quad (4.6.1)$$

Le théorème suivant est alors vérifié.

Théorème 4.6.1.2. *Soit une séquence partielle de tâches terminée à un instant t après la complétion d'une tâche k . Pour tout couple de tâches (i, j) , et par rapport au coût des avances et des retards ($\sum_{i=1}^n (\max(0, d_i - c_i) + \max(0, c_i - d_i))$) des tâches, $i \preceq j$ si et seulement si*

$$ETRS(i, j, k, t) \leq ETRS(j, i, k, t) \quad (4.6.2)$$

La preuve de ce théorème est analogue à celle du théorème 4.5.1.2. Il suffira pour cela de remplacer dans la démonstration $R_i = \max(r_i, t)$ (resp. $R_j = \max(r_j, t)$) par $R_i^k = \max(r_i, t + s_{ki})$ (resp. $R_j^k = \max(r_j, t + s_{kj})$), et à la fin d'une tâche (i ou j), tenir compte des temps logistiques s_{ij} et s_{ji} . Nous n'allons donc pas reprendre cette démonstration.

Remarque 4.6.1.3. *Si le coût est exprimé comme une somme pondérée des pénalités d'avance et de retard, avec l'avance pondérée par W_e et le retard pondéré par W_t , alors dans ce cas le théorème 4.6.1.2 est toujours valable en remplaçant la fonction ETR par la suivante*

$$ETRSW(i, j, t) = W_e \cdot |\max(r_i, t + s_{ki}) + p_i - d_i| + W_t \cdot |\max(\max(r_i, t + s_{ki}) + p_i + s_{ij}, r_j) + p_j - d_j| \quad (4.6.3)$$

Tous les développements ultérieurs sont alors toujours valables.

Remarque 4.6.1.4. *Tout comme dans le cas où les temps logistiques sont faibles, si le coût minimal pour la tâche i est atteint sur une tranche horaire $[e_i, d_i]$ plutôt qu'à une date précise d_i , on peut remplacer la fonction ETRS par $\max(e_i - (R_i^k + p_i), 0) + \max((R_i^k + p_i) - d_i, 0) + \max(e_j - (\max(R_i^k + p_i + s_{ij}, r_j) + p_j), 0) + \max((\max(R_i^k + p_i + s_{ij}, r_j) + p_j) - d_j, 0)$ où $R_i^k = \max(r_i, t + s_{ki})$ et $R_j^k = \max(r_j, t + s_{kj})$. Cela dépend alors de la tâche k qui est complétée. Et les outils développés ici peuvent toujours être utilisés.*

Le problème de recherche d'une borne inférieure à ce problème reste un problème ouvert. Le principe de l'utilisation des fonctions ETRS et ETRSW est donné dans la sous-section ci-après.

4.6.2 Algorithme de base

La procédure de base utilise une ressource.

Algorithme 4.6.2.1. Principe de base du ETRS et du ETRWS : AlgoOF-ETRS, AlgoOF-ETRWS
Cas d'une ressource ($M = 1$)

A tout instant t où la ressource est disponible ou libérée après la complétion d'une tâche k , les étapes suivantes permettent la prise de décision. La tâche la plus prioritaire est sélectionnée à la fin de ces étapes et est ordonnancée. Toutes ces étapes sont répétées jusqu'à la dernière tâche.

1. Construire l'ensemble $S = \{\text{toutes les tâches non encore ordonnancées}\}$.
2. Calculer $ETRS(i,j,k,t)$ (ou $ETRSW(i,j,k,t)$ selon le cas) pour toutes les paires de tâches $\{i,j\}$ dans S .
3. Calculer la matrice de dominance Ω dans S .
4. Calculer le vecteur des forces (f_i) dans S .
5. Déterminer le sous-ensemble S' des tâches ayant la plus grande force dans S .
 $S \leftarrow S'$.
 Tant que $\text{card}(S) > 1$, aller à l'étape 4.

Remarque 4.6.2.2. *Cet algorithme 4.6.2.1 peut aussi, comme les autres, être adapté dans le cas de ressources parallèles avec un nombre M de ressources supérieur ou égale à deux.*

La complexité de l'algorithme AlgoOF-ETRS (resp. AlgoOF-ETRWS) est donnée dans la proposition suivante. La démonstration de cette proposition est analogue est celle de l'algorithme AlgoOF-PRTWF dans la proposition 3.4.3.3 du chapitre 3.

Proposition 4.6.2.3. *L'algorithme AlgoOF-ETRS (resp. AlgoOF-ETRWS) basé sur la fonction ETRS (resp. ETRWS), la matrice de dominance et le vecteur des forces a une complexité en $O(n^3)$ si n est le nombre de tâches à ordonnancer.*

Nous allons illustrer cette procédure avant de passer à son utilisation pour le problème (\mathcal{P}_{32}).

4.6.3 Exemple 4.4 illustratif de la procédure AlgoOF-ETRS

Ici aussi, nous n'illustrons que la procédure avec la fonction ETRS, celle avec la fonction ETRSW se ramenant à la même chose (aux coefficients W_e et W_t près). Nous considérons pour cette illustration l'instance de l'exemple 4.2 avec la fonction FTRS, sachant bien sûr qu'ici le coût des avances et des retards des tâches est considéré, au lieu du coût des temps de réponse et des retards dans l'exemple 4.2. L'instance de cet exemple 4.4 est donc la même que sur les tableaux 4.6 et 4.7.

La matrice des valeurs de la fonction ETRS, la matrice de dominance et le vecteur des forces des tâches à l'instant $t = 0$ (la tâche fictive 0 étant complétée à cet instant) sont représentés respectivement sur les tableaux 4.17, 4.18 et 4.19.

Après la sélection de la tâche 1 à cette date, les autres décisions dans l'exemple 4.4 sont représentées sur le tableau 4.20.

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	0,38	0,10	2,44	1,98
2	4,84	-	2,80	5,36	6,16
3	3,01	2,21	-	2,72	3,23
4	4,53	2,67	3,63	-	6,33
5	2,03	0,29	0,25	2,00	-

Table 4.17 – Valeurs de la fonction ETRS à $t = 0$ dans l'exemple 4.4

$\downarrow i, j \rightarrow$	1	2	3	4	5
1	-	1	1	1	1
2	0	-	0	0	0
3	0	1	-	1	1
4	0	1	0	-	0
5	0	1	0	1	-

Table 4.18 – Matrice de dominance à $t = 0$ dans l'exemple 4.4

Tâche i	1	2	3	4	5
Force f_i	4	0	3	1	2

Table 4.19 – Vecteur des forces des tâches à $t = 0$ dans l'exemple 4.4

Instant de décision (t)	1,90	4,41	4,72	5,66
Tâche complétée	1	5	3	4
Ensemble des tâches	{2, 3, 4, 5}	{2, 3, 4}	{2, 4}	{2}
Tâche sélectionnée	5	3	4	2
Date de début (après set-up)	2,38	4,50	5,15	5,95

Table 4.20 – Autres décisions dans l'exemple 4.4 pour la fonction ETRS

La séquence d'exécution des tâches dans l'exemple 4.4 est alors comme sur le tableau 4.21, avec un coût moyen combiné d'avances et de retards des tâches égal à 2,042.

Ordre d'exécution	1	2	3	4	5
Tâche	1	5	3	4	2
Date de début	1,14	2,38	4,50	5,15	5,95
Date de fin	1,90	4,41	4,72	5,66	8,89

Table 4.21 – Séquence d'exécution des tâches de l'exemple 4.4

Nous allons maintenant repasser au problème (\mathcal{P}_{32}) pour la procédure temps-réel dans l'optique de sa résolution.

4.6.4 Algorithme en temps-réel du problème (\mathcal{P}_{32})

Dans la procédure temps-réel du problème (\mathcal{P}_{32}) dans la suite de cette section, nous parlerons indifféremment, tout comme dans les précédents problèmes, de sélection de tâches de maintenance sur une entité ou de sélection de l'entité elle-même. Dans cette procédure temps-réel, nous utiliserons,

en plus de la fonction *ETRS* et du théorème 4.6.1.2, la définition de tâches urgentes introduite dans 3.3.1.4 au chapitre 3.

Sur la base de tout ce qui précède, la procédure suivante est proposée pour une prise de décision en temps-réel dans le problème (\mathcal{P}_{32}).

Algorithme 4.6.4.1. Procédures temps-réel OL-MTSA-ETPS

Au début de l'horizon de planification \mathcal{H} ($t = 0$):

Déterminer les premières dates critiques et les premiers délais de maintenance préventive sur chaque entité du système comme définis dans la position du problème.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 5.*
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
4. *Appliquer la procédure décrite dans 4.4.2.1 (AlgoOF-ETRS et AlgoOF-ETRWS) à l'ensemble S , pour sélectionner la tâche à ordonnancer.
Déterminer sa date de fin.
Déterminer la date de la prochaine tâche et la date de maintenance optimale sur l'entité \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^3 à sa date de fin.
Aller à l'étape 1.*
5. *Fin.*

Cette procédure a une complexité qui est dans la proposition ci-dessous.

Proposition 4.6.4.2. Complexité

Si N^l est le nombre total de tâches de maintenance entreprises sur l'horizon \mathcal{H} , alors la complexité de la procédure OL-MTSA-ETPS est de l'ordre de $N^l \left(\sum_{k=1}^N N_k\right)^2$.

La justification de cette proposition est analogue à celle de la complexité de la procédure OL-MTSA-W dans le chapitre 3.

Quelques résultats d'expérimentation sur la méthode sont dans la sous-section qui va suivre.

4.6.5 Résultats numériques

Expérimentations

Les expérimentations sont conduites sur le protocole d'expérimentation dans la sous-section 4.3.6 avec à présent les données relatives aux temps logistiques entre les sites. La configuration du système en nombre d'entités et en nombre de ressources est la même que dans la section 4.4. Les résultats sont représentés sur la figure 4.5. Sur les figures 4.5(a), 4.5(b), 4.5(c) 4.5(d) et 4.5(f), les paramètres ont les mêmes significations que dans la section 4.5 et sur la figure 4.5(e), ils ont les mêmes significations que dans la section 4.4.

Analyse des résultats

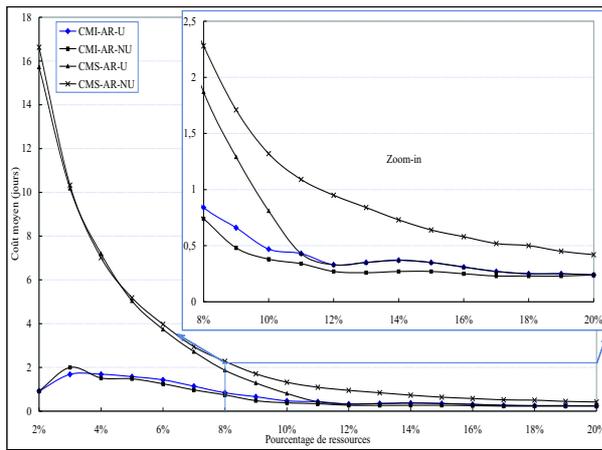
Les observations sur les figures 4.5(b), 4.5(e) et 4.5(f) sont analogues à celles de la sous-section 3.3.4 du chapitre 3. Pour les figures 4.5(a) et 4.5(c), elles sont analogues à celles de la section précédente. Sur la figure 4.5(d), la différence entre la composante des coûts moyens CMI (seulement sur les tâches effectivement entreprises sur l'horizon \mathcal{H}) due aux retards et celle due aux avances, est beaucoup plus flagrante que dans la section précédente avec des temps logistiques faibles. Cela est due aux temps logistiques des ressources entre les sites, amplifiant les retards des tâches qu'elles entreprennent. Mais à part cela, les observations de cette figure sont analogues à celles sur sa correspondante de la section précédente.

4.7 Conclusion

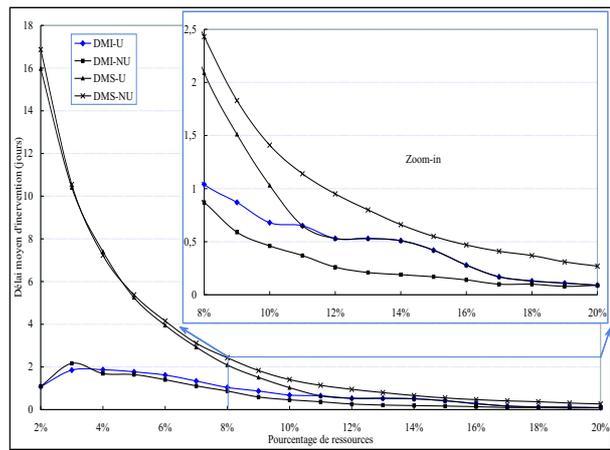
Dans ce chapitre, nous avons abordé les problèmes de décision en temps-réel dans un processus de maintenance dans un système distribué et ce, dans un registre complémentaire à celui du chapitre 3. Les problèmes considérés dans ce chapitre impliquent des contraintes de délais sur les tâches de maintenance. C'est le cas lorsqu'il y a, en plus des coûts d'états critiques traités dans le chapitre 3, des coûts dus à des retards des tâches de maintenance par rapport à des délais liés au comportement du système et aux exigences de production. Par ailleurs, certaines situations exigent que les tâches de maintenance soient de préférence entreprises à des dates bien déterminés afin d'engendrer un moindre coût. Cela peut être une date lié à la mise ou remise en service des entités ou bien sur une plage horaire de tolérance. Des déviations de ces dates (ou des périodes de tolérance) engendrent soit des coûts d'avance des tâches de maintenance, soit des coûts de retard de ces tâches.

Nous avons considéré ces types de problèmes avec, d'une part, des coûts combinés des temps critiques et des retards de maintenance et, d'autre part, des coûts combinés des avances et des retards des tâches de maintenance. Pour chacun de ces types de problèmes, nous avons travaillé suivant deux configurations du système distribué :

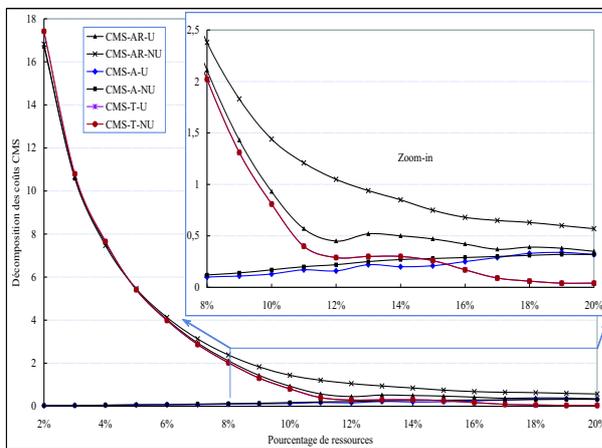
- les différents sites du système sont dans une configuration telle que les temps logistiques entre eux soient faibles et donc assimilés à 0 (ce peut être le cas lorsque les sites du système sont regroupés sur un même site géographique),



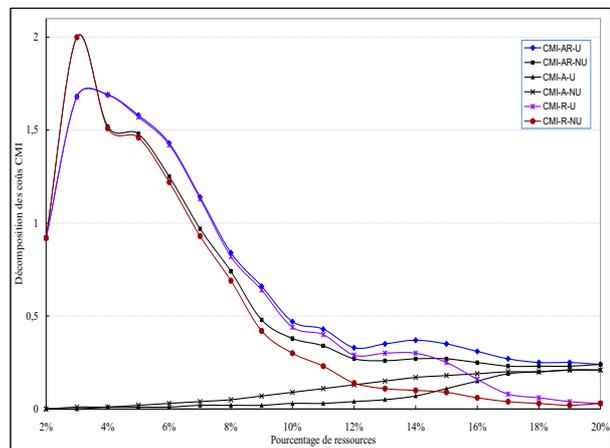
(a) Coût moyen d'avances et retards



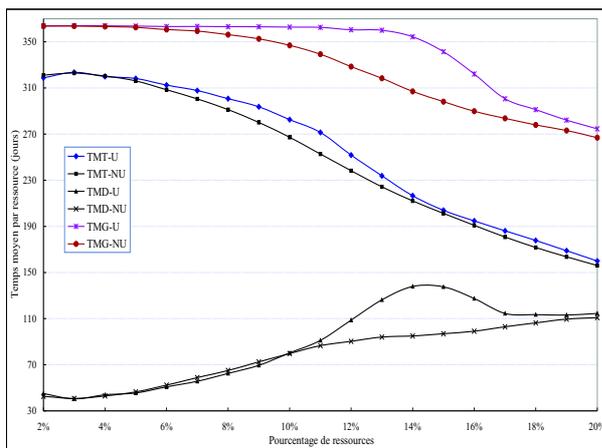
(b) Délai moyen d'intervention



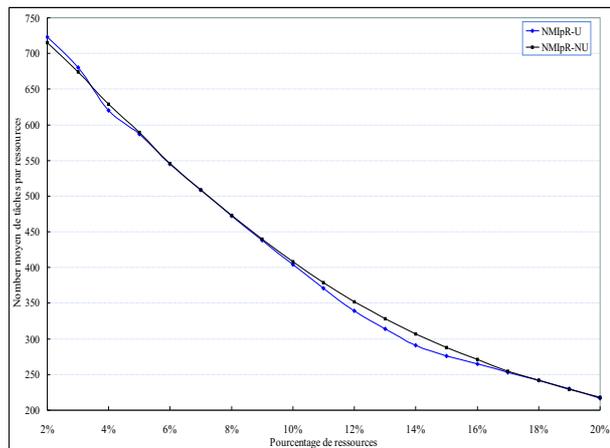
(c) Décomposition des coûts CMS



(d) Décomposition des coûts CMI



(e) Temps moyen d'activité par ressource



(f) Nombre moyen de tâches par ressource

Figure 4.5 – Résultats des coûts d'avance et de retards avec des temps logistiques importants

- la configuration du système (qui est alors très distribué géographiquement) est telle que les temps logistiques entre les sites sont non négligeables et aucune hypothèse réductrice n'est possible à ce niveau.

Dans chacune de ces configurations, nous avons considéré des critères de décision localement optimaux sur des problèmes localement statiques pour les besoins de la prise de décision en temps-réel dans le processus de maintenance. Ces critères dépendent des aspects de coût en question et en général, les fonctions qui les définissent ne se rapportent pas à une seule tâche de maintenance à la fois et elles ne définissent pas des relations d'ordre total dans l'ensemble des tâches à une date de décision. Leurs utilisations se sont alors faites sur la base de la matrice de dominance et du vecteur des forces tâches introduits dans le chapitre 3. Des exemples simples illustrent l'utilisation de chacune de ces règles avant de passer à leur utilisation concrète dans les problèmes temps-réels.

L'utilisation de ces règles dans des heuristiques a des complexités polynômiales faibles et elles peuvent être utilisées dans la résolution de problèmes de grandes tailles. Par ailleurs, ces règles peuvent être utilisées dans des méthodes de résolution du type *branch and bound* de problèmes statiques, dans lesquels les différentes tâches ont des dates d'occurrence différentes avec des coûts se rapportant à ceux traités dans ce chapitre.

Dans tout ce chapitre comme dans le chapitre 3, les tâches de maintenance sur les entités sont supposées avoir des durées d'exécution fixes connues qui sont leur *MTTR*. Les événements critiques considérés sur les entités sont réduits au franchissement de seuils critiques de disponibilités de ces entités, se rapportant donc à des processus de type maintenance préventive par rapport à l'évolution de la disponibilité des entités du système. Les procédures décrites dans ce chapitre peuvent tout aussi être adaptées dans des simulations aux cas où les durées des tâches de maintenance sont plutôt aléatoires et où des pannes fortuites interviennent dans le fonctionnement du système. Nous reviendrons sur ces points dans le chapitre 5.

Chapitre 5

Organisation de la Maintenance et Simulation Monte-Carlo

5.1 Introduction

Dans les précédents chapitres, nous avons abordé le processus de maintenance sous un aspect partiellement déterministe. Les disponibilités des différentes entités du système considéré sont aléatoires et suivent des lois supposées connues. Seule la maintenance préventive était prise en compte à des dates qui sont planifiées dès la mise en service ou la remise en service d'une entité. Les durées des tâches de maintenance sont supposées connues pour chaque entité nécessitant une intervention de maintenance préventive. Sur la base de toutes ces considérations et avec des approches de coûts différents, nous avons pu établir des règles pour l'aide à la décision en temps-réel dans le processus de maintenance.

Dans le but d'établir la robustesse des approches d'aide à la décision proposées dans les précédents chapitres sous un aspect partiellement déterministe, nous allons considérer dans le présent chapitre le processus de maintenance sous un angle entièrement stochastique. Non seulement les disponibilités des entités sont aléatoires mais toutes les autres caractéristiques le sont aussi. Les actions de maintenance préventives seront planifiées dès la mise en service du système ou la remise en service des entités ; des pannes fortuites et non planifiées seront aussi générées sur les entités lors du fonctionnement du système. Ces pannes aussi bien que les actions de maintenance préventive seront considérées en temps-réel . Les durées des tâches de maintenance préventive et curative sur les entités du système seront aléatoires et liées à leur MTTR.

5.2 Position du problème

Nous n'allons pas reprendre ici tous les détails sur les problèmes abordés dans ce document. Le lecteur peut pour cela voir la position des problèmes spécifiés dans les chapitres 3 et 4. Nous rappelons simplement que les problèmes abordés sont des problèmes de gestion temps-réel des processus de maintenance sur un système distribué. Ce système comprend plusieurs sites distribués

composés d'entités réparables. Les ressources de maintenance utilisées sont partagées par tous les sites du système. Les différentes autres hypothèses dans les chapitres 3 et 4 sont valables dans celui-ci. En particulier, les tâches de maintenance sont supposées être parfaites et sont exécutées sans préemption. Dans les approches proposées ici, les ressources de maintenance sont supposées de même performance, et utilisées en parallèle. Cette hypothèse, comme nous le verrons dans la conclusion générale, n'est pas restrictive et des extensions des approches peuvent être faites pour des cas de ressources de différentes classes.

Les différents critères de coûts considérés jusqu'à présent sont de trois types principaux à savoir (se référer aux chapitres 3 et 4 pour les notations) :

- les coûts dus à des états critiques des entités du système

$$C_{tot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} w_{ik} (c_{m,ik} - r_{m,ik}) \right) \quad (5.2.1)$$

ou en particulier

$$C'_{tot}(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} (c_{m,ik} - r_{m,ik}) \right) \quad (5.2.2)$$

- les coûts combinés des états critiques et des retards des tâches de maintenance

$$C_2(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} ((c_{m,ik} - r_{m,ik}) + \max(0, c_{m,ik} - d_{m,ik})) \right) \quad (5.2.3)$$

ou de façon plus générale sous forme pondérée

$$C_2^w(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/r_{m,ik} \in \mathcal{H}} (W_f \cdot (c_{m,ik} - r_{m,ik}) + W_t \cdot \max(0, c_{m,ik} - d_{m,ik})) \right) \quad (5.2.4)$$

- les coûts dus à des avances et des retards de maintenance par rapport à des délais requis

$$C_3(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/d_{m,ik} \in \mathcal{H}} (\max(0, c_{m,ik} - d_{m,ik}) + \max(0, d_{m,ik} - c_{m,ik})) \right) \quad (5.2.5)$$

ou sous forme pondérée

$$C_3^w(\mathcal{H}) = \sum_{k=1}^N \sum_{i=1}^{N_k} \left(\sum_{m/d_{m,ik} \in \mathcal{H}} (W_e \cdot \max(0, c_{m,ik} - d_{m,ik}) + W_t \cdot \max(0, d_{m,ik} - c_{m,ik})) \right) \quad (5.2.6)$$

Jusqu'ici, les événements critiques concernant les entités se rapportent principalement à la maintenance préventive. Ces événements sont liés aux franchissements de seuils de disponibilité. Par ailleurs, les durées des tâches de maintenance sur une entité \mathcal{E}_{ik} sont supposées constantes sur l'horizon de planification \mathcal{H} et sont égales au MTTR $\left(\frac{1}{\mu_{ik}}\right)$ de cette entité.

Dans ce chapitre, les franchissements de seuils de disponibilité menant alors à de la maintenance préventive ne sont plus les seuls événements critiques sur les entités du système. D'autres événements se rapportant à des pannes fortuites peuvent se produire dans le système. Les durées

des tâches de maintenance sur une entité \mathcal{E}_{ik} ne sont plus supposées constantes sur l'horizon de planification \mathcal{H} . Certes, le taux de réparation de chaque entité \mathcal{E}_{ik} est toujours supposé constant sur l'horizon \mathcal{H} , mais les durées des tâches sont maintenant des variables aléatoires liées à ce taux de réparation. On peut donc faire le lien et voir que la durée d'une tâche de maintenance sur une entité \mathcal{E}_{ik} est alors une variable aléatoire suivant la loi exponentielle de paramètre μ_{ik} sur l'horizon de planification \mathcal{H} .

Une entité mise en service ou remise en service à une date T peut être sujette à des pannes dont l'occurrence est liée au taux de défaillance de cette entité. Le taux de défaillance λ_{ik} d'une entité \mathcal{E}_{ik} peut être constante sur l'horizon \mathcal{H} . Dans ce cas, la durée depuis la mise ou la remise en service de l'entité jusqu'à l'occurrence d'une panne est une variable aléatoire suivant la loi exponentielle de paramètre λ_{ik} . Sinon, ce taux de défaillance peut avoir les caractéristiques d'une loi de Weibull telle que λ_{ik} soit défini par

$$\begin{aligned} \lambda_{ik} : [\max(0, \gamma), +\infty[&\longrightarrow \mathbb{R}_+ \\ t &\longmapsto \lambda(t) = \beta \frac{(t - \gamma)^{(\beta-1)}}{\sigma^\beta} \end{aligned} \quad (5.2.7)$$

avec $\sigma \in \mathbb{R}_+^*$, $\beta \in \mathbb{R}_+^*$ et $\gamma \in \mathbb{R}$ (voir dans l'appendice B pour les détails) et où le paramètre d'échelle σ , le paramètre de forme β et le paramètre de décalage γ , sont respectivement remplacés par les correspondants individuels σ_{ik} , β_{ik} et γ_{ik} de ces paramètres sur l'entité \mathcal{E}_{ik} . Dans ce cas, la loi suivie par la durée de fonctionnement avant défaillance dépend de la valeur du paramètre de forme β_{ik} .

En somme, pour une entité \mathcal{E}_{ik} mise ou remise en service, elle tombera en panne après une durée, ou alors si elle fonctionne "assez" longtemps sans panne, elle franchira son seuil critique α_{ik}^1 de disponibilité. L'occurrence d'une panne fortuite ou le franchissement des seuils critiques de disponibilité est un événement critique pour l'entité en question. En fait, la durée jusqu'à cet événement critique pour l'entité \mathcal{E}_{ik} est $\tau_{ik} + \varepsilon_{ik}$ où les détails sur τ_{ik} peuvent être lus dans les chapitres 3 et 4 et où ε_{ik} est une variable aléatoire. Cette dernière variable peut prendre une valeur négative si une panne survient avant le franchissement du seuil critique de disponibilité. Mais cette durée reste à τ_{ik} si la tâche de maintenance préventive est entreprise avant l'occurrence d'une panne sur l'entité \mathcal{E}_{ik} . Si la durée de fonctionnement depuis la remise en service de l'entité \mathcal{E}_{ik} dépasse τ_{ik} (ce qui veut dire que $A_{ik} \leq \alpha_{ik}$) mais que cette entité n'est pas en panne, elle est alors dans un état critique de fonctionnement avec une forte probabilité de défaillance. Les problèmes sont alors en relation avec l'un des problèmes \mathcal{P}_1 , \mathcal{P}_2 , ou \mathcal{P}_3 pour lesquels le lecteur peut se référer au chapitre 3 ou 4 pour des détails. Nous reviendrons individuellement sur chacun de ces problèmes dans le contexte actuel.

Pour tenir compte de la présence de ces durées aléatoires dans le fonctionnement du système, nous utilisons la simulation Monte-Carlo qui prend en compte ces aspects aléatoires du problème. Avant de reconsidérer les problèmes en question, rappelons brièvement ce qu'est la simulation Monte-Carlo.

5.3 L'outil de simulation Monte-Carlo

La méthode Monte-Carlo est un outil de simulation des processus partiellement ou entièrement aléatoires. Elle a été au départ développée pour la recherche atomique mais son principe était utilisé bien avant le vingtième siècle. Elle est basée sur l'usage des nombres aléatoires pour générer des données et simuler des processus. Elle utilise principalement le rapprochement entre des estimations statistiques et la réalité. L'historique de la méthode Monte-Carlo, ni même sa théorie, n'étant l'objet de ce document, nous inviterons le lecteur à consulter les différents ouvrages qui sont nombreuses sur le sujet.

Néanmoins, nous rappellerons que son formalisme a été établi dans les années 1940. Dans les années 1950 et 1960, elle a été largement testée dans plusieurs domaines d'ingénierie, chimie, physique, recherche opérationnelle, statistiques. Pour plus de détails sur son évolution, le lecteur peut voir notamment dans l'ouvrage de Fishman [63], de Mooney [112], de Rubinstein [132]. Pour d'amples détails sur ses concepts, on peut se référer aussi à l'ouvrage de Fox [64], de Gentle [72], Hellekalek et Larcher [80]. On se référera aussi, pour plus d'exemples sur les applications, à l'ouvrage de Dubi [60], de Niederreiter et Jau-Shyong Shiue [121], en plus de l'ouvrage de Rubinstein [132]. Nous laisserons donc au lecteur le choix de trouver les différentes possibilités intéressantes dans l'usage de l'outil Monte-Carlo et les adaptations très variées qu'on peut faire de cet outil. Nous aborderons directement l'usage que nous avons fait de la méthode Monte-Carlo dans les problèmes abordés dans ce document et son adaptation dans les méthodes d'aide à la décision développées dans les précédents chapitres.

Les problèmes énumérés plus haut seront considérés successivement. Nous ne reviendrons pas ici sur les différentes généralités de ces problèmes. Nous adapterons les différentes approches dans les chapitres précédents aux différentes configurations des problèmes \mathcal{P}_1 , \mathcal{P}_2 et \mathcal{P}_3 . La section suivante concerne le problème \mathcal{P}_1 .

5.4 Algorithmes de simulation pour les problèmes \mathcal{P}_1

Dans cette section, nous utiliserons une adaptation des règles de priorité utilisées dans le chapitre 3 à savoir les fonctions PRTF, PRTWF, PRTFSS, PRTFST et PRTWFST, et les différents outils qui ont servi dans les algorithmes temps-réel dans le chapitre 3. Ici, des pannes fortuites sont générées tout au long du fonctionnement de l'entité et les durées des tâches de maintenance sont aussi aléatoires.

5.4.1 Simulation pour le problème \mathcal{P}_{11}

Pour les généralités sur ce problème, se référer à la section 3.3 du chapitre 3.

Algorithme 5.4.1.1. Procédures de simulation pour le problème \mathcal{P}_{11}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($\text{Expo}(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autres que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.
4. Déterminer dans l'ensemble S l'entité à traiter en sélectionnant celle dont la tâche donne la plus petite valeur de la fonction $PRTF$ (voir dans le chapitre 3) pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $\text{Expo}(\lambda_{ik})$.
Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
Aller à l'étape 1.
6. Fin.

La sous-section suivante présente l'algorithme d'adaptation pour le problème \mathcal{P}_{12} dans le contexte actuel.

5.4.2 Simulation pour le problème \mathcal{P}_{12}

Pour les généralités sur ce problème, se référer à la section 3.4 du chapitre 3.

Algorithme 5.4.2.1. Procédure de simulation pour le problème \mathcal{P}_{12}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($\text{Expo}(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autres que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.
4. Appliquer la procédure décrite dans l'algorithme 3.4.3.1 (AlgoOF-PRTWF) dans le chapitre 3 à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $\text{Expo}(\lambda_{ik})$.
Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
Aller à l'étape 1.
6. Fin.

Avec l'introduction des temps logistiques considérables entre les sites, nous considérons maintenant l'adaptation de la procédure temps-réel pour le problème \mathcal{P}_{13} dans le contexte actuel.

5.4.3 Simulation pour le problème \mathcal{P}_{13}

Pour les généralités sur ce problème, se référer à la section 3.5 du chapitre 3.

Algorithme 5.4.3.1. Procédures de simulation pour le problème \mathcal{P}_{13}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($\text{Expo}(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autres que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.
4. Appliquer, suivant que les temps logistiques entre les sites sont symétriques ou non, la procédure décrite dans l'algorithme 3.5.2.1 (AlgoOF-PRTFST) ou dans l'algorithme 3.5.2.2 (AlgoOF-PRTFSS) du chapitre 3 à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $\text{Expo}(\lambda_{ik})$.
Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} question en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
Aller à l'étape 1.
6. Fin.

Finalement, la sous-section suivante présente l'adaptation de la procédure temps-réel de la dernière configuration et de la plus générale du problème \mathcal{P}_1 au contexte actuel.

5.4.4 Simulation pour le problème \mathcal{P}_{14}

Pour les généralités sur ce problème, se référer à la section 3.6 du chapitre 3.

Algorithme 5.4.4.1. Procédure de simulation pour le problème \mathcal{P}_{14}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($\text{Expo}(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autres que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.
2. Déterminer le sous-ensemble \mathcal{U}_t du système.
3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $\mathcal{S} = \mathcal{U}_t$.
Sinon $\mathcal{S} = \{\text{toutes les entités de tous les sites}\}$.
4. Appliquer la procédure décrite dans l'algorithme 3.6.2.1 (AlgoOF-PRTWFST) du chapitre 3 à l'ensemble \mathcal{S} , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.
5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $\text{Expo}(\lambda_{ik})$.
Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} question en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
Aller à l'étape 1.
6. Fin.

La section suivante revient au deuxième type de problème \mathcal{P}_2 .

5.5 Algorithmes de simulation pour les problèmes \mathcal{P}_2

Nous utiliserons ici une adaptation des règles de priorité utilisées pour le problème \mathcal{P}_2 dans le chapitre 4 à savoir les fonctions FTR, FTRW, FTRS et FTRSW et les différents outils qui ont servi à leur utilisation dans les procédures temps-réel pour le problème \mathcal{P}_2 . Puisque les états critiques sont en jeu dans ce problème \mathcal{P}_2 aussi, des pannes fortuites (engendrant aussi des états critiques) seront générées lors du fonctionnement, en plus des durées d'exécution aléatoire.

5.5.1 Simulation pour le problème \mathcal{P}_{21}

Pour les généralités sur ce problème, se référer à la section 4.3 du chapitre 4.

Algorithme 5.5.1.1. Procédure de simulation pour le problème \mathcal{P}_{21}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($Expo(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autre que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $Expo(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

- 1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.*
- 2. Déterminer le sous-ensemble \mathcal{U}_t du système.*
- 3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
- 4. Appliquer la procédure décrite dans 4.3.2.1 (AlgoOF-FTR et AlgoOF-FTRW) (dans le chapitre 4) à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.*
- 5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $Expo(\lambda_{ik})$.
Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans*

la position du problème) à sa date de fin.

Déterminer la date de la prochaine tâche et le délai sur l'entité \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^2 à sa date de fin.

Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.

Aller à l'étape 1.

6. Fin.

La configuration de ce problème avec des temps logistiques importants entre les sites est considérée dans la sous-section qui suit. L'algorithme présente une adaptation de la procédure en temps-réel pour le problème \mathcal{P}_{22} dans le contexte actuel.

5.5.2 Simulation pour le problème \mathcal{P}_{22}

Pour les généralités sur ce problème, se référer à la section 4.4 du chapitre 4.

Algorithme 5.5.2.1. Procédure de simulation pour le problème \mathcal{P}_{22}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Générer les dates de pannes estimées sur chaque entité dans le système. Sur une entité \mathcal{E}_{ik} (entité i du site S_k), cela se fait suivant la loi exponentielle de moyenne $\frac{1}{\lambda_{ik}}$ ($\text{Expo}(\lambda_{ik})$).

Déterminer les premières dates critiques sur chaque entité du système (autres que les dates de pannes) en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

- 1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource). La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à ce instant.
Relever le site S_k où se trouve la ressource sélectionnée.
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.*
- 2. Déterminer le sous-ensemble \mathcal{U}_t du système.*
- 3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
- 4. Appliquer la procédure décrite dans 4.4.2.1 (AlgoOF-FTRS et AlgoOF-FTRWS) (dans le chapitre 4) à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
Déterminer sa date de fin.*

5. Générer la date de panne probable suivante de l'entité \mathcal{E}_{ik} en question suivant la loi $Expo(\lambda_{ik})$.
 Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
 Déterminer la date de la prochaine tâche préventive et le délai sur l'entité \mathcal{E}_{ik} en question en ajoutant respectivement sa quantité τ_{ik} et τ_{ik}^2 à sa date de fin.
 Générer la durée d'exécution de tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $Expo(\mu_{ik})$.
 Aller à l'étape 1.
6. Fin.

Le second type de problème considéré avec des contraintes de délais sur les tâches de maintenance est l'objet de la section ci-après.

5.6 Algorithmes de simulation pour les problèmes \mathcal{P}_3

Pour les procédures dans cette section, nous allons considérer les règles de priorité utilisées dans le problème \mathcal{P}_3 à savoir les fonctions ETR, ETRW, ETRS et ETRSW. Des adaptations de ces règles seront utilisées avec les différents outils dont nous avons fait usage dans les procédures temps-réel pour le problème \mathcal{P}_3 dans le chapitre 4. Les états critiques n'étant pas explicitement l'enjeu ici, et puisque les pannes induisent plutôt des coûts d'états critiques, les pannes ne seront pas générées dans ces procédures. Mais les durées aléatoires sur la base des MTTRs des entités seront générées.

5.6.1 Simulation pour le problème \mathcal{P}_{31}

Pour les généralités sur ce problème, se référer à la section 4.5 du chapitre 4.

Algorithme 5.6.1.1. Procédure de simulation pour le problème \mathcal{P}_{31}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Déterminer les premières dates critiques sur chaque entité du système en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $Expo(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource).
 La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.

- Relever le site S_k où se trouve la ressource sélectionnée.*
Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.
2. *Déterminer le sous-ensemble \mathcal{U}_t du système.*
 3. *Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
 Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
 4. *Appliquer la procédure décrite dans 4.5.2.1 (AlgoOF-ETR et AlgoOF-ETRW) (dans le chapitre 4) à l'ensemble S , pour sélectionner la tâche à ordonnancer (nous supposons que cette entité est \mathcal{E}_{ik}).
 Déterminer sa date de fin.*
 5. *Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
 Déterminer la date suivante de maintenance de coût minimal de \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik}^3 à sa date de fin.
 Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
 Aller à l'étape 1.*
 6. *Fin.*

Il faut noter que comme observé dans les résultats d'expérimentation sur le problème \mathcal{P}_{31} et \mathcal{P}_{32} dans le chapitre 4, la procédure de sélection de la tâche à ordonnancer donne ici aussi un meilleur coût lorsqu'elle est utilisée avec la notion d'urgence introduite au paravent.

5.6.2 Simulation pour le problème \mathcal{P}_{32}

Pour les généralités sur ce problème, se référer à la section 4.6 du chapitre 4.

Algorithme 5.6.2.1. Procédure de simulation pour le problème \mathcal{P}_{32}

Au début de l'horizon de planification \mathcal{H} ($t = 0$) :

Déterminer les premières dates critiques sur chaque entité du système en utilisant la durée τ_{ik} telle que définie dans le chapitre 3 dans la position du problème.

Générer les durées d'exécution des tâches de maintenance à venir sur chaque entité du système. Cette génération se fait sur l'entité \mathcal{E}_{ik} suivant la loi $\text{Expo}(\mu_{ik})$.

Les décisions sont ensuite prises suivant les étapes ci-dessous.

1. *Considérer l'instant de décision t (date de disponibilité au plus tôt d'une ressource).
 La ressource sélectionnée est celle de plus petit indice parmi les ressources disponibles à cet instant.
 Relever le site S_k où se trouve la ressource sélectionnée.*

- Si $t < \mathcal{H}$, alors aller à l'étape 2. Sinon aller à l'étape 6.*
- 2. Déterminer le sous-ensemble \mathcal{U}_t du système.*
 - 3. Si $\text{card}(\mathcal{U}_t) \geq 1$, alors l'ensemble de recherche $S = \mathcal{U}_t$.
Sinon $S = \{\text{toutes les entités de tous les sites}\}$.*
 - 4. Appliquer la procédure décrite dans 4.4.2.1 (AlgoOF-ETRS et AlgoOF-ETRWS) (dans le chapitre 4) à l'ensemble S , pour sélectionner la tâche à ordonnancer.
Déterminer sa date de fin.*
 - 5. Déterminer la date d'occurrence de la tâche de maintenance préventive suivante sur l'entité \mathcal{E}_{ik} en ajoutant sa quantité τ_{ik} (telle que définie dans le chapitre 3 dans la position du problème) à sa date de fin.
Déterminer la date suivante de maintenance de coût minimal de \mathcal{E}_{ik} en question en ajoutant sa quantité τ_{ik}^3 à sa date de fin.
Générer la durée d'exécution de la tâche de maintenance à venir sur l'entité \mathcal{E}_{ik} avec la loi $\text{Expo}(\mu_{ik})$.
Aller à l'étape 1.*
 - 6. Fin.*

Nous regroupons dans la section ci-après les expérimentations sur ces différentes procédures. Pour ces expérimentations, et dans l'optique d'analyser ces résultats par rapports à ceux des chapitres précédents, nous utiliseront les mêmes instances de données que dans ces chapitres.

5.7 Résultats numériques

5.7.1 Protocole d'expérimentation

Comme dit ci-dessus, les expérimentations se font sur les données du chapitre 3 et du chapitre 4. Ces données sont décrites dans la sous-section 3.3.3 du chapitre 3 et la sous-section 4.3.6 du chapitre 4.

Afin de bien situer les règles d'aide à la décision dans des environnements partiellement déterministes par rapport à des environnement entièrement aléatoires au moyen de la simulation Monte-Carlo, les expérimentations, dans tous les cas de figure qui se sont présentés, ont été conduites chacune suivant trois protocoles différents. Dans un premier temps, les systèmes sont considérés dans les mêmes cas de figure que dans les précédents chapitres. Cela veut dire que les disponibilités des entités du système sont aléatoires, mais seule la maintenance préventive est considérée dans ce cas. En plus, les durées d'exécution des tâches de maintenance dépendent bien sûr de chaque entité mais sont constantes pour chaque tâche sur l'horizon de planification \mathcal{H} . Dans un second temps, le système est toujours considéré avec uniquement de la maintenance préventive mais les durées d'exécution pour chaque entité sont aléatoires et sont liées au $MTTR$ connu de chaque entité. Dans un troisième temps, en plus des durées d'exécution qui sont aléatoires, des pannes fortuites sont générées suivant la procédure des algorithmes décrits plus haut.

Ainsi, sur les figures représentant les résultats, les légendes sont en général formées de trois

parties et sont sous la forme A_B_C. La partie A désigne le critère de coût représenté par chaque figure ou les autres paramètres représentés dans les chapitres 3 et 4. Le lecteur pourra se référer à ces chapitres pour la signification des paramètres de la partie A. La partie B quant à elle est représentée comme DP, RP ou RPF. DP (Deterministic Processing) signifie que le paramètre est relevé dans des conditions où la maintenance préventive est seule prise en compte et les durées des tâches sur chaque entité sont déterministes. RP (Random Processing) représente les paramètres dans le cas où seule la maintenance préventive est prise en compte mais, contrairement à DP, les durées d'exécution sont aléatoires. Finalement, RPF (Random Processing and Failure) dans la partie B désigne les relevés des paramètres avec aussi bien la maintenance préventive que la maintenance curative suite à des pannes fortuites aléatoirement générées tout au long de l'horizon de planification \mathcal{H} . Dans la partie C, nous avons la notation U (notion d'urgence utilisée) et NU (dans le cas contraire).

Compte tenu du grand nombre de courbes de chaque type et pour des raisons de lisibilité des figures, quelques courbes représentées ensemble dans les chapitres précédents seront séparées sur les sous-figures différentes qui vont suivre et regroupées avec leurs homologues. Nous utilisons toujours le terme coût moyen pour désigner la valeur du critère de coût considéré mais ce coût moyen ne signifie pas la même chose dans tous les problèmes.

5.7.2 Résultats du problème \mathcal{P}_{11}

Les courbes de cette configuration sont sur la figure 5.1. Le coût ici est relatif aux états critiques correspondant à des temps de réponse avec des tâches de maintenance non pondérées.

5.7.3 Résultats du problème \mathcal{P}_{12}

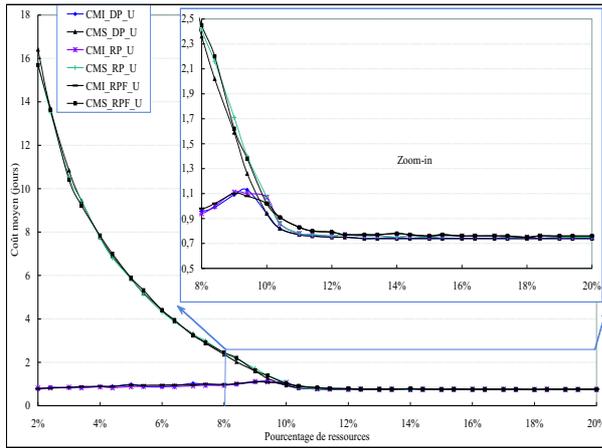
Les courbes de cette configuration sont sur la figure 5.2. Le coût ici est relatif aux états critiques correspondant à des temps de réponse pondérés avec des tâches de maintenance pondérées.

5.7.4 Résultats du problème \mathcal{P}_{13}

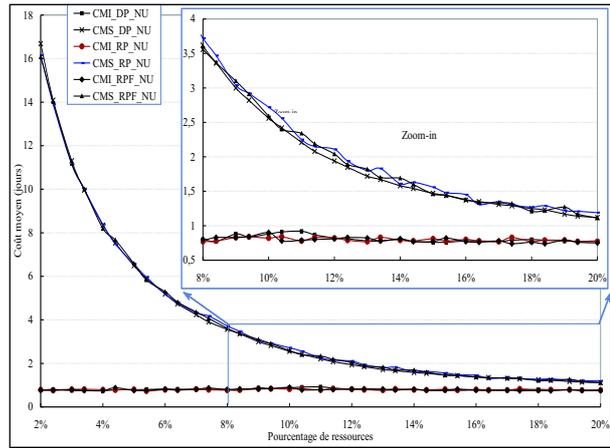
Les courbes de cette configuration sont sur la figure 5.3. Le coût ici est relatif aux états critiques correspondant à des temps de réponse avec des tâches de maintenance non pondérées.

5.7.5 Résultats du problème \mathcal{P}_{14}

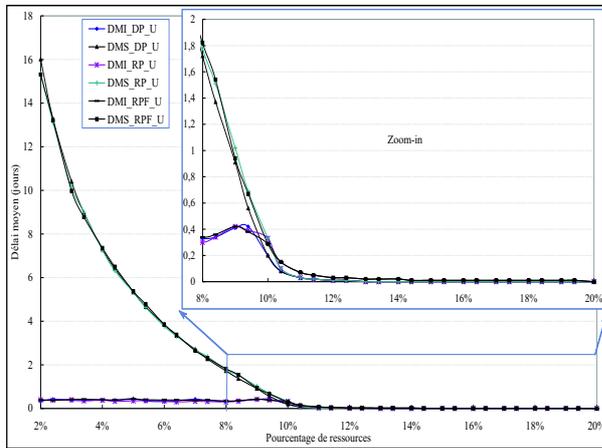
Les courbes de cette configuration sont sur la figure 5.4. Le coût ici est relatif aux états critiques correspondant à des temps de réponse pondérés avec des tâches de maintenance pondérées.



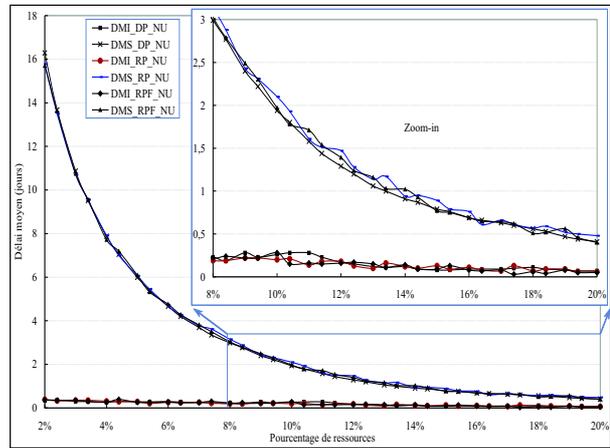
(a) Coût moyen de maintenance (avec urgence)



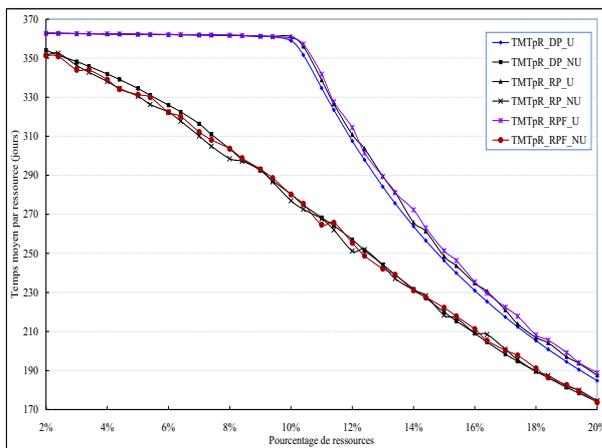
(b) Coût moyen de maintenance (sans urgence)



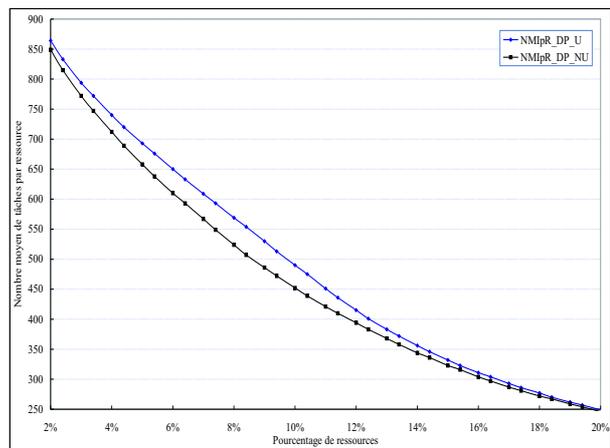
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

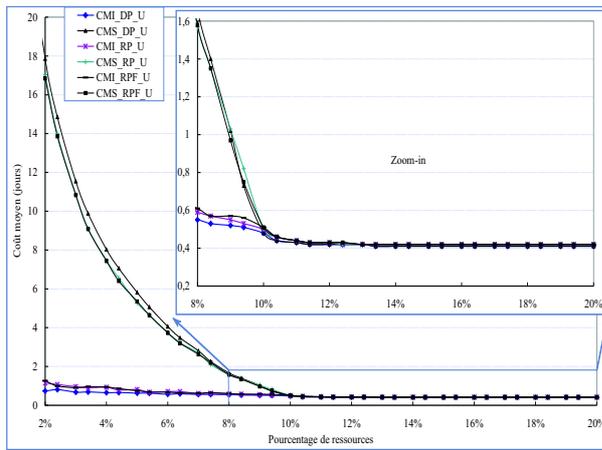


(e) Temps moyen d'activité par ressource

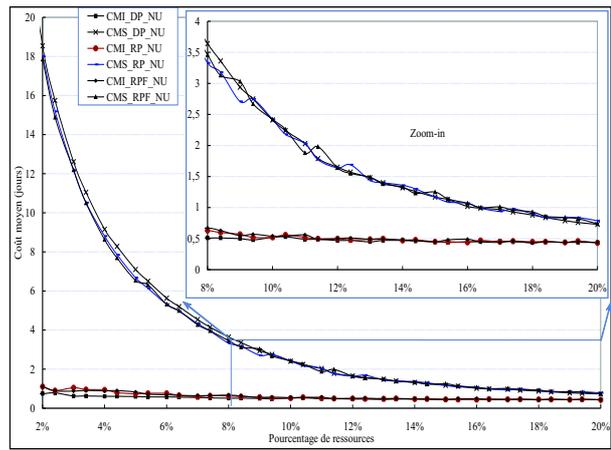


(f) Nombre moyen d'interventions par ressource

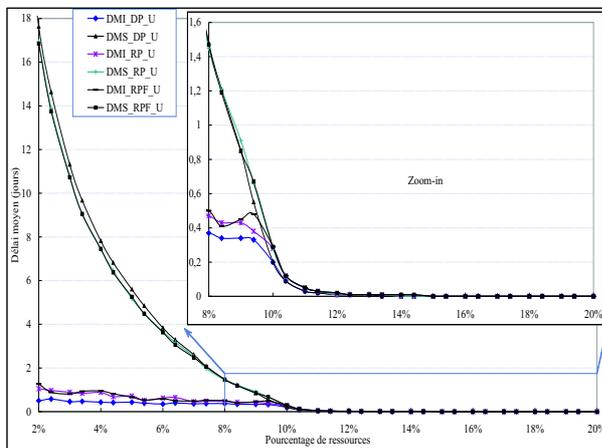
Figure 5.1 – Résultats de simulation pour le problème \mathcal{P}_{11}



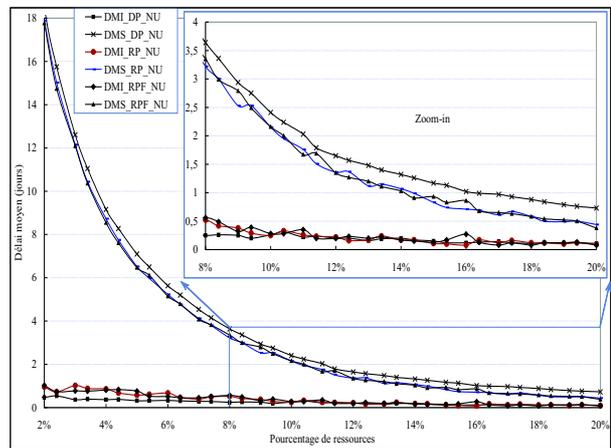
(a) Coût moyen de maintenance (avec urgence)



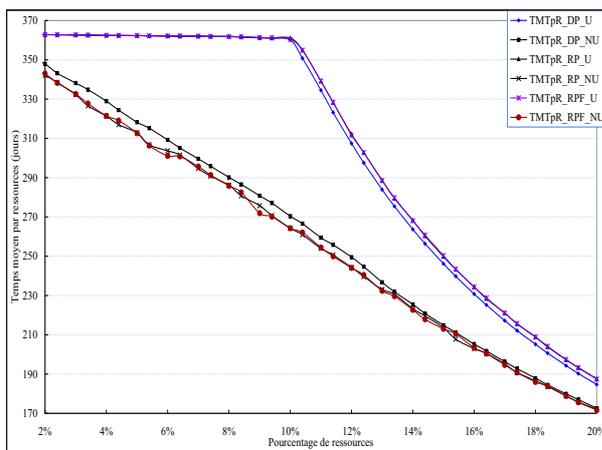
(b) Coût moyen de maintenance (sans urgence)



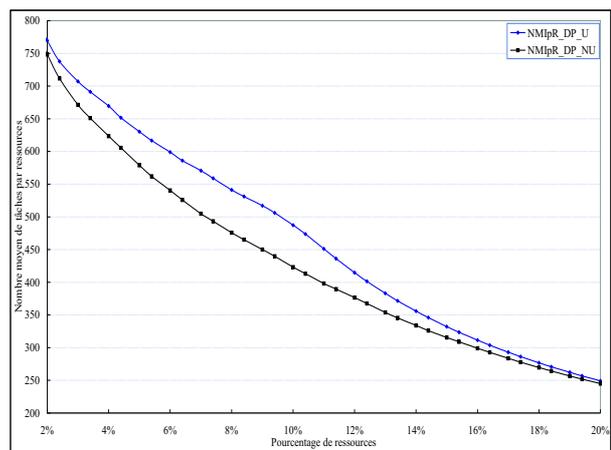
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

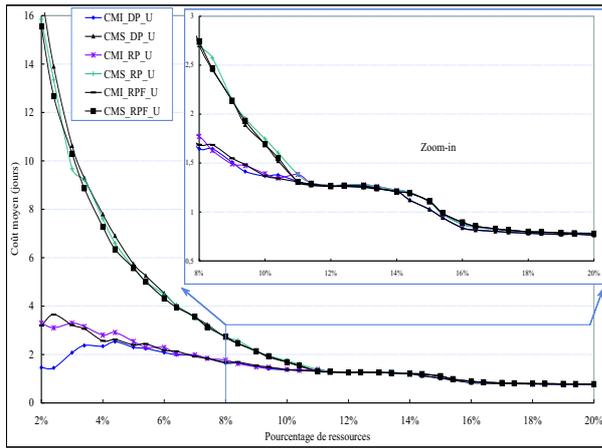


(e) Temps moyen d'activité par ressource

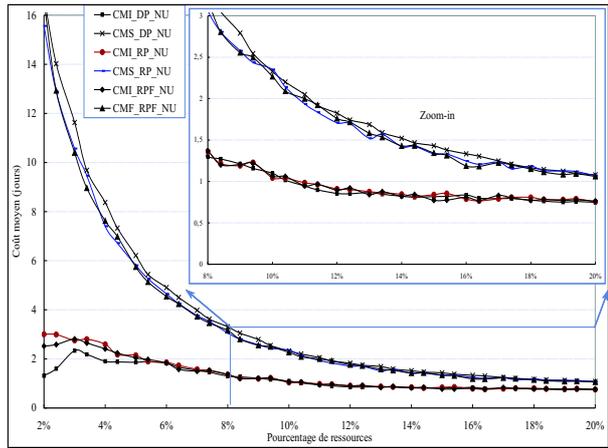


(f) Nombre moyen d'interventions par ressource

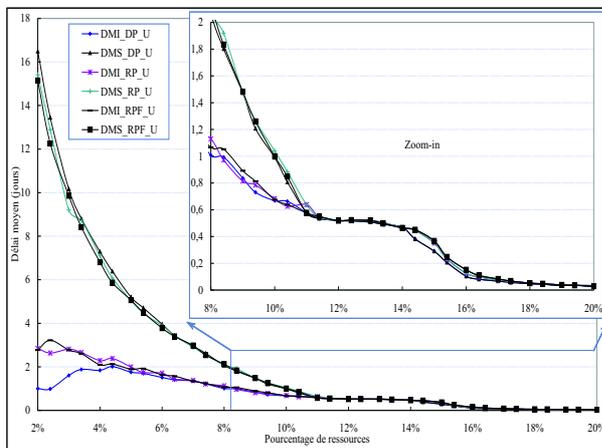
Figure 5.2 – Résultats de simulation pour le problème \mathcal{P}_{12}



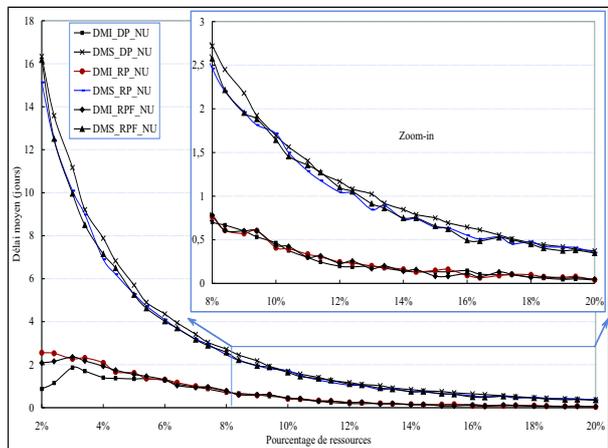
(a) Coût moyen de maintenance (avec urgence)



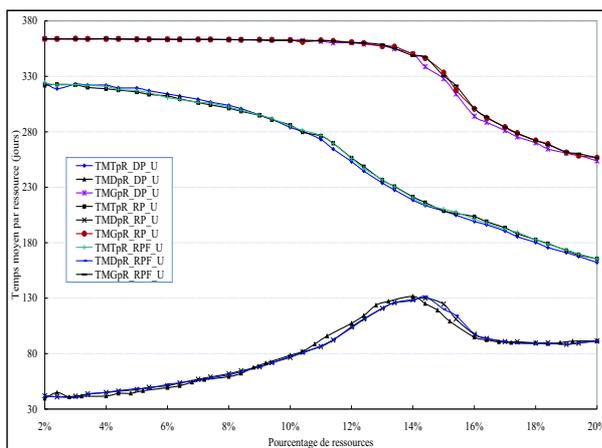
(b) Coût moyen de maintenance (sans urgence)



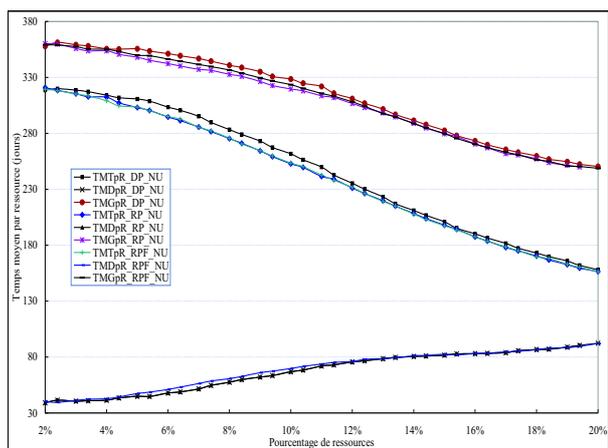
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

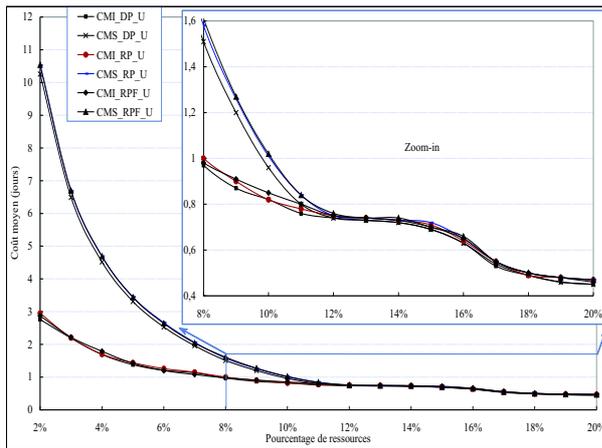


(e) Temps moyen d'activité par ressource (avec urgence)

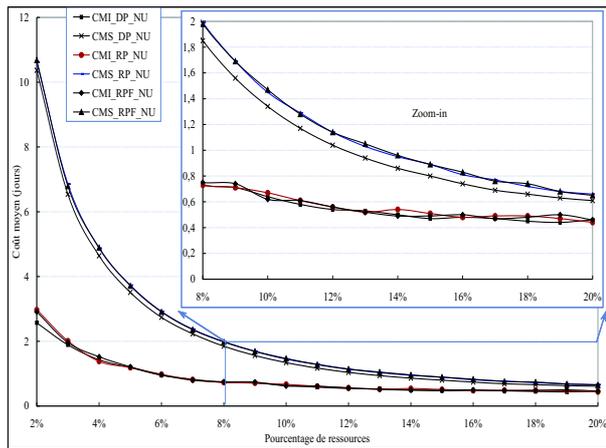


(f) Temps moyen d'activité par ressource (sans urgence)

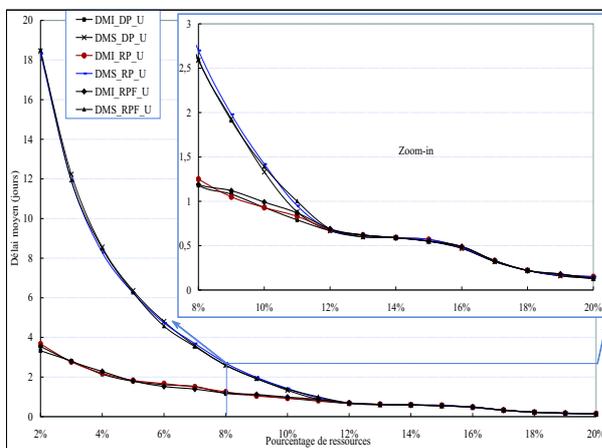
Figure 5.3 – Résultats de simulation pour le problème \mathcal{P}_{13}



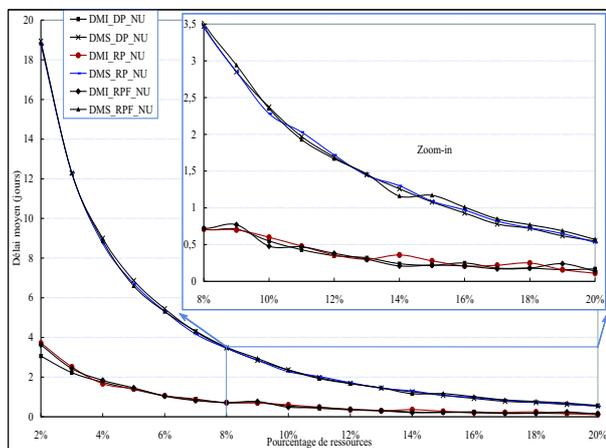
(a) Coût moyen de maintenance (avec urgence)



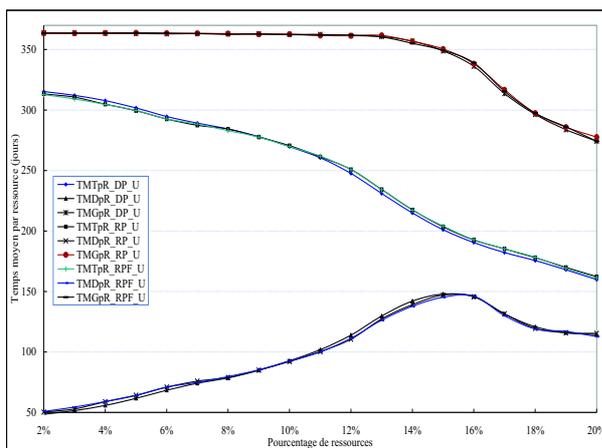
(b) Coût moyen de maintenance (sans urgence)



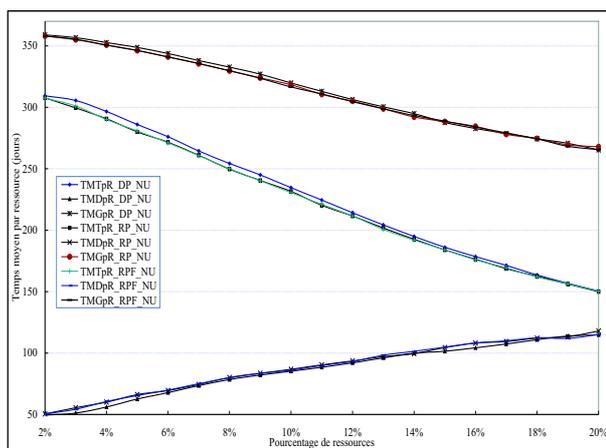
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)



(e) Temps moyen d'activité par ressource (avec urgence)



(f) Temps moyen d'activité par ressource (sans urgence)

Figure 5.4 – Résultats de simulation pour le problème \mathcal{P}_{14}

5.7.6 Résultats du problème \mathcal{P}_{21}

Les courbes de cette configuration sont sur la figure 5.5. Le coût moyen est relatif aux états critiques et aux retard des tâches de maintenance. Nous présentons une décomposition des coûts sur l'ensemble du système comme dans le chapitre 4 mais pour le coût relevé uniquement sur les tâches de maintenance entreprises, nous ne présentons que le coût composé des états critiques et des retards des tâches de maintenance. Les décompositions ont les mêmes évolutions que dans le chapitre 4. À cause du nombre important de courbes sur chaque sous-figure, nous ne les représentons pas.

5.7.7 Résultats du problème \mathcal{P}_{22}

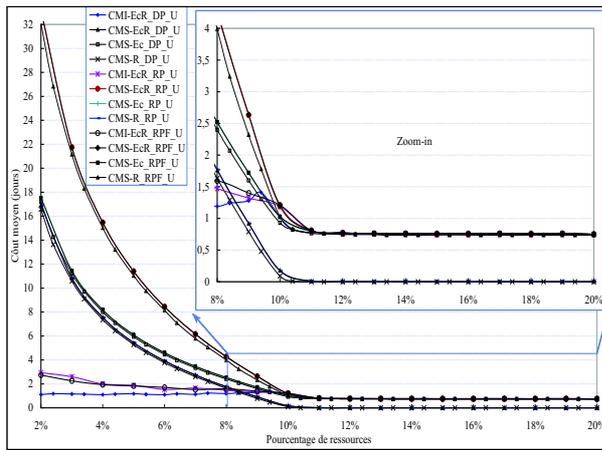
Les courbes de cette configuration sont sur la figure 5.6. Le coût moyen est relatif aux états critiques et aux retard des tâches de maintenance. Comme dans la sous-section précédente, nous présentons une décomposition des coûts sur l'ensemble du système comme dans le chapitre 4. Pour les mêmes raisons que dans la sous-section précédente, pour le coût relevé uniquement sur les tâches de maintenance entreprises, nous ne présentons que le coût composé des états critiques et des retards des tâches de maintenance.

5.7.8 Résultats du problème \mathcal{P}_{31}

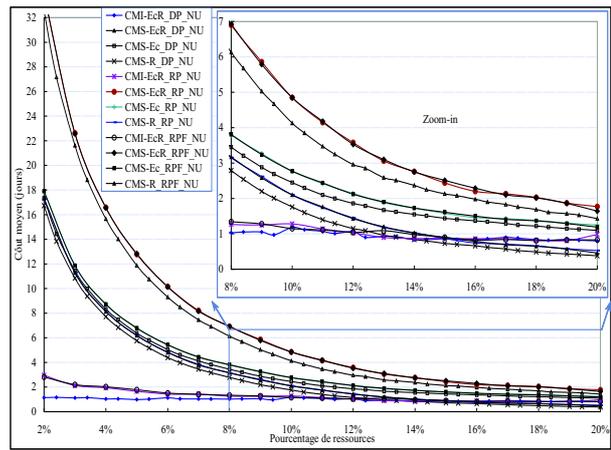
Les courbes de cette configuration sont sur la figure 5.7. Le coût moyen est relatif au déficit et aux fréquences élevées de maintenance. Il correspond aux avances et retards des tâches de maintenance. Ici, nous ne représentons pas les décompositions de ce coût, puisque nous avons vu dans les résultats du chapitre 4 que la plus grande partie de ce coût dans les expérimentations, est due aux retards de maintenance.

5.7.9 Résultats du problème \mathcal{P}_{32}

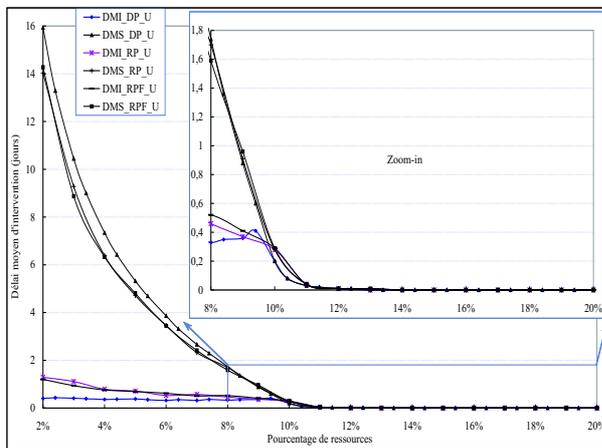
Les courbes de cette configuration sont sur la figure 5.8. Le coût moyen est comme dans la sous-section précédente, relatif au déficit et aux fréquences élevées de maintenance. Il est correspond aux avances et retards des tâches de maintenance. Pour les mêmes raisons que dans la sous-section précédente, nous ne représentons pas ici non plus les décompositions de ce coût.



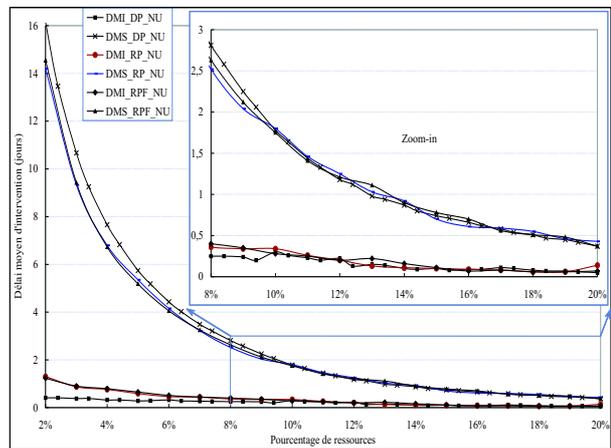
(a) Coût moyen de maintenance (avec urgence)



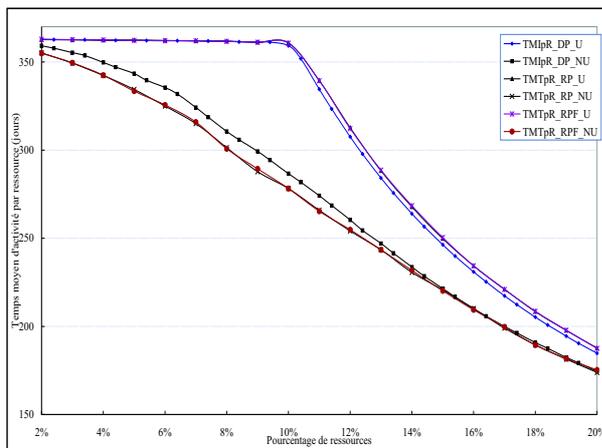
(b) Coût moyen de maintenance (sans urgence)



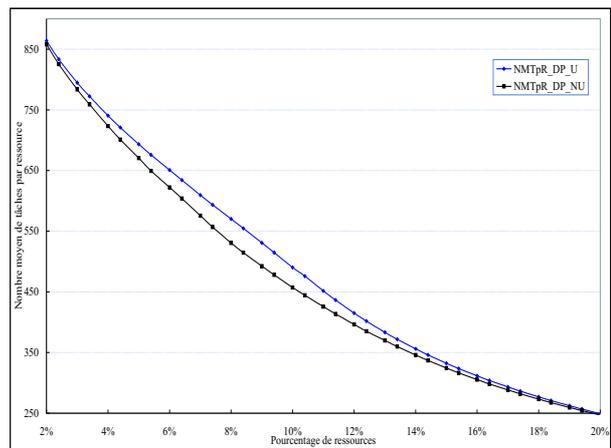
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

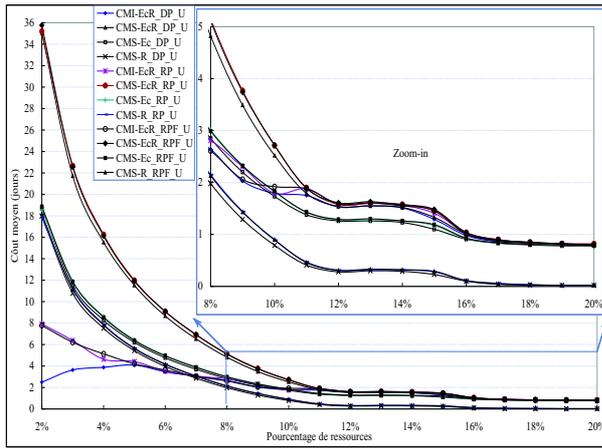


(e) Temps moyen d'activité par ressource

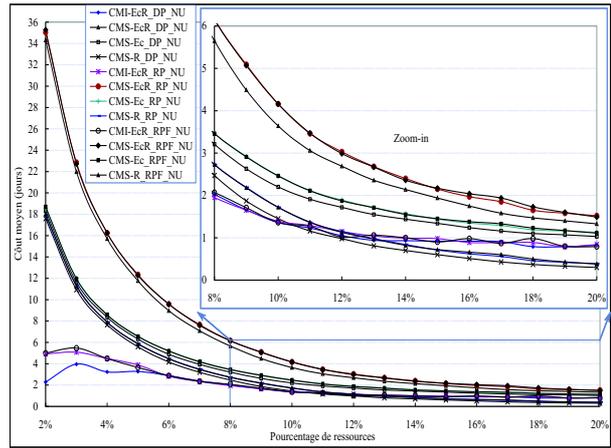


(f) Nombre moyen d'interventions par ressource

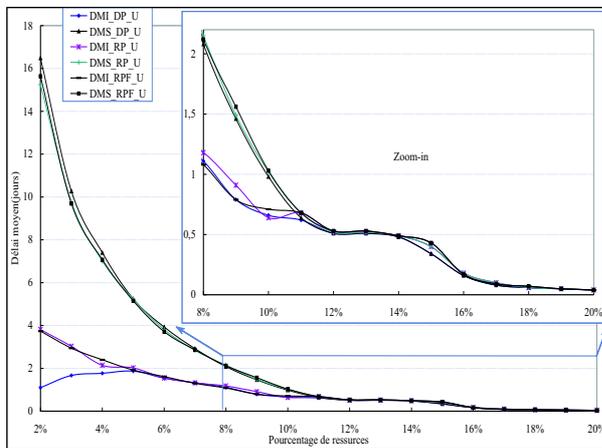
Figure 5.5 – Résultats de simulation pour le problème \mathcal{P}_{21}



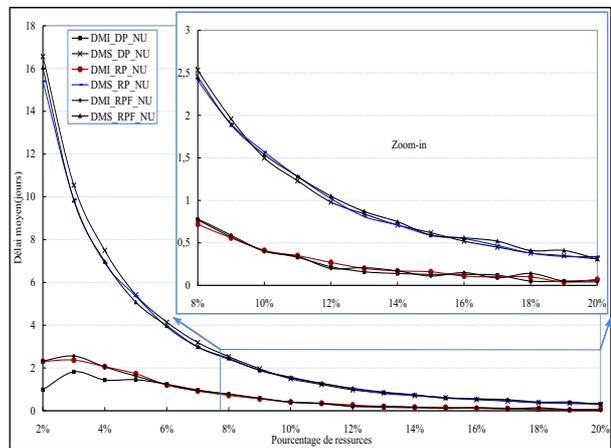
(a) Coût moyen de maintenance (avec urgence)



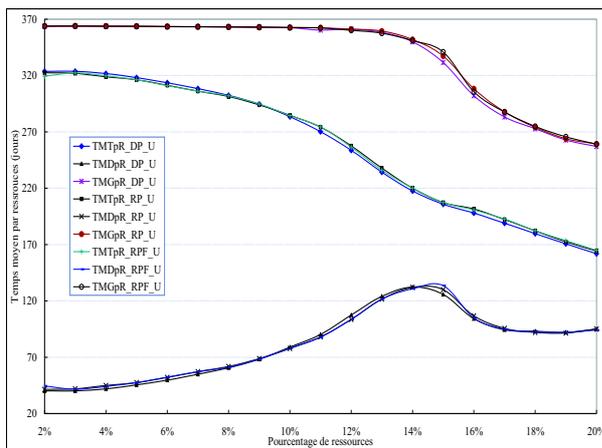
(b) Coût moyen de maintenance (sans urgence)



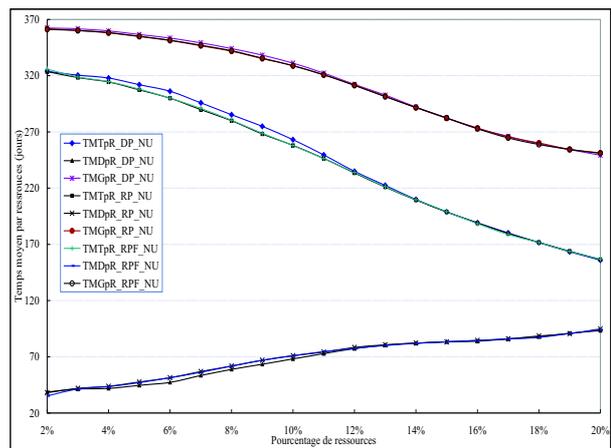
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

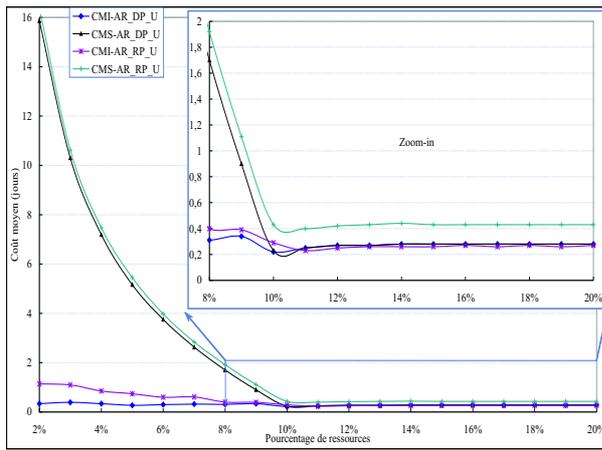


(e) Temps moyen d'activité par ressource (avec urgence)

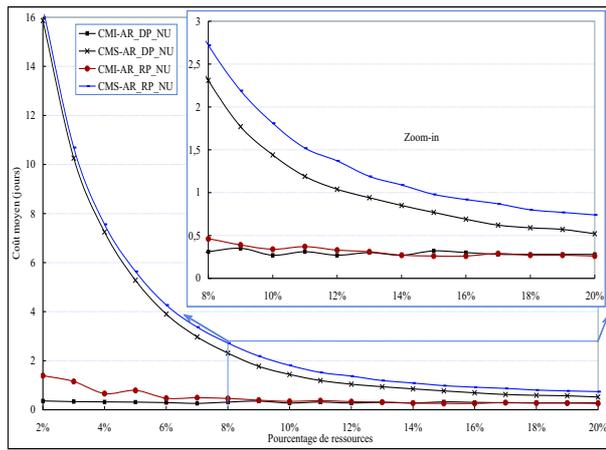


(f) Temps moyen d'activité par ressource (sans urgence)

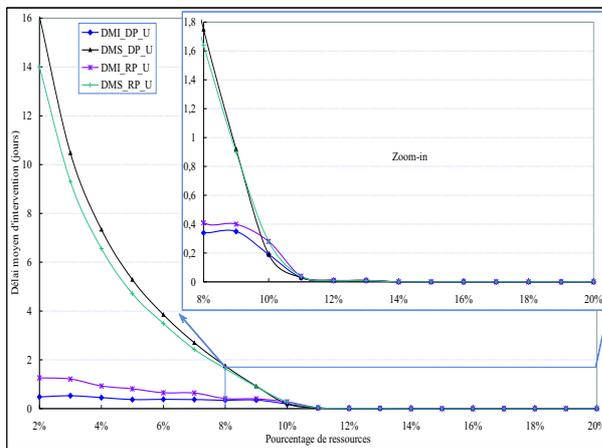
Figure 5.6 – Résultats de simulation pour le problème \mathcal{P}_{22}



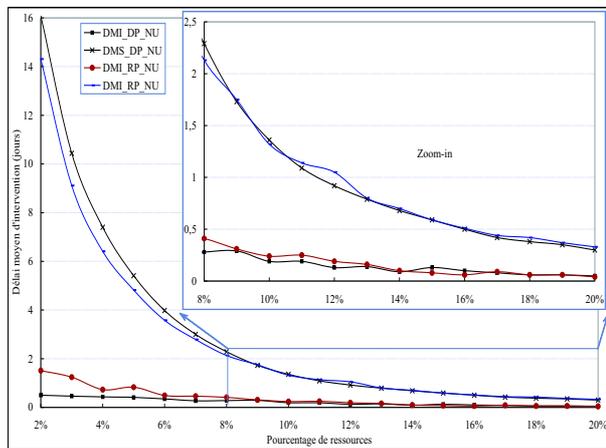
(a) Coût moyen de maintenance (avec urgence)



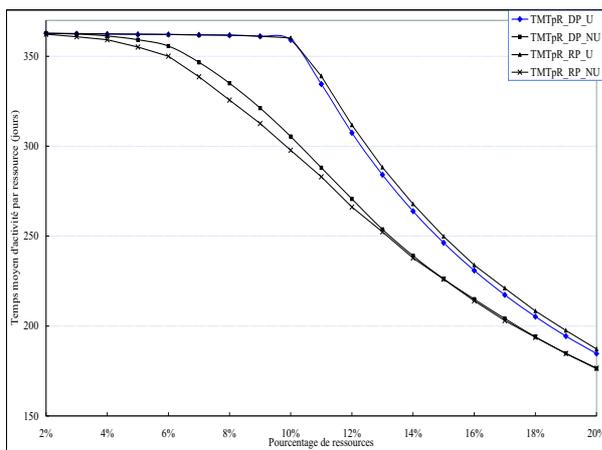
(b) Coût moyen de maintenance (sans urgence)



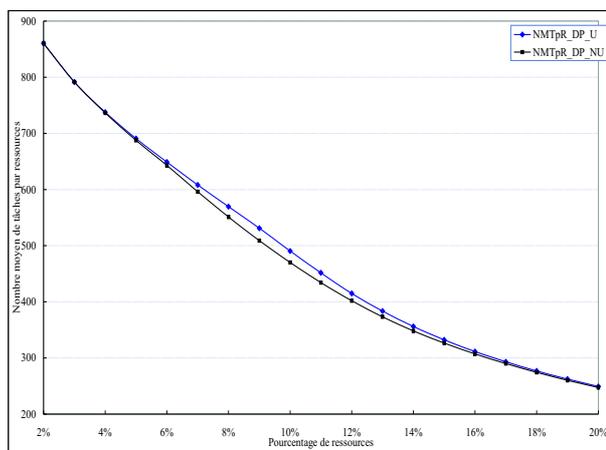
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)

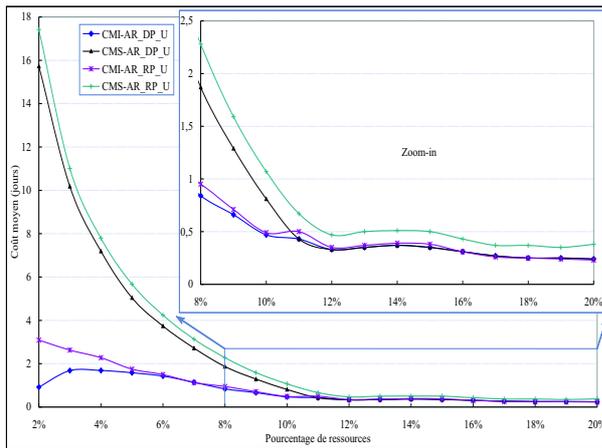


(e) Temps moyen d'activité par ressource

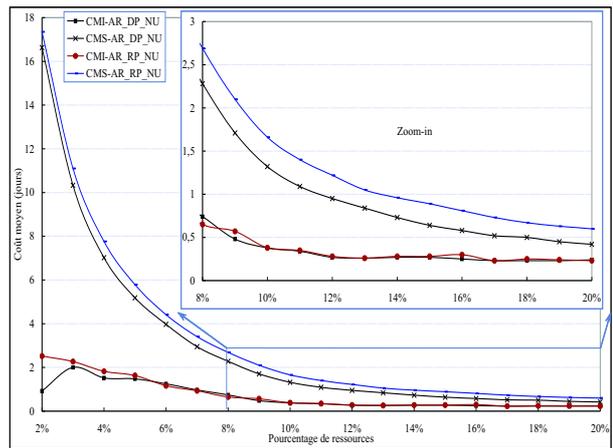


(f) Nombre moyen d'interventions par ressource

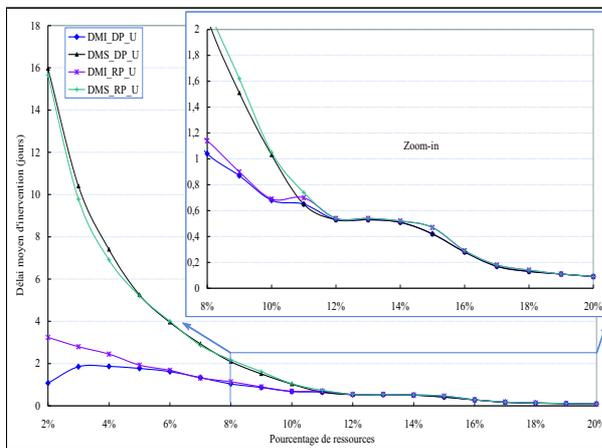
Figure 5.7 – Résultats de simulation pour le problème \mathcal{P}_{31}



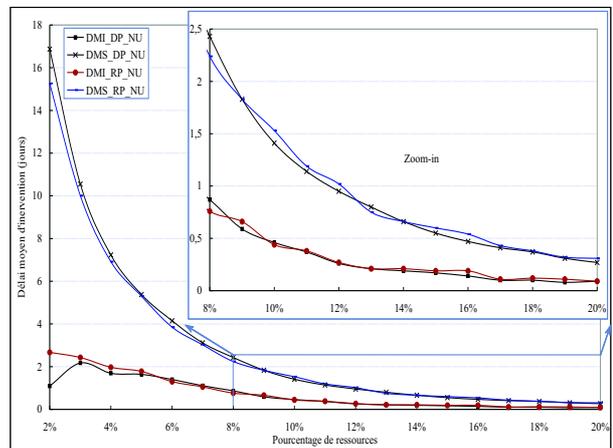
(a) Coût moyen de maintenance (avec urgence)



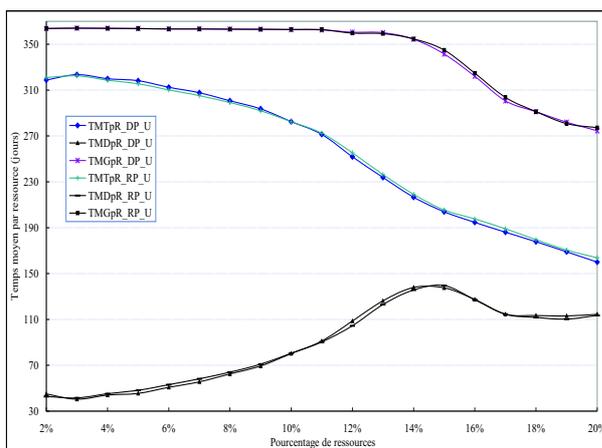
(b) Coût moyen de maintenance (sans urgence)



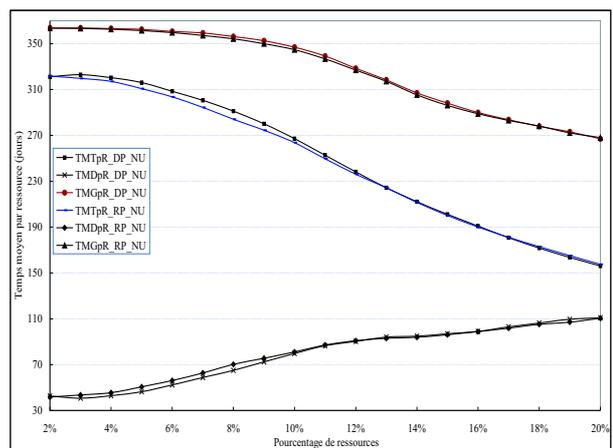
(c) Délai moyen d'intervention (avec urgence)



(d) Délai moyen d'intervention (sans urgence)



(e) Temps moyen d'activité par ressource (avec urgence)



(f) Temps moyen d'activité par ressource (sans urgence)

Figure 5.8 – Résultats de simulation pour le problème \mathcal{P}_{32}

5.8 Observations et conclusion

Dans ce chapitre, nous avons intégré les modèles de prise de décision développés dans les chapitres précédents dans des algorithmes de simulation. Ces simulations viennent compléter les hypothèses utilisées dans les chapitres 3 et 4 où les durées d'exécution des tâches de maintenance sont constantes sur l'horizon de planification pour chacune des entités du système étudié et où il n'y avait pas de pannes dans le fonctionnement du système. Dans ces simulations, les durées d'exécution des tâches de maintenance sur chaque entité du système sont aléatoires et des pannes aléatoires sont générées au cours du fonctionnement du système.

Pour chacun des critères de coûts moyens étudiés, toutes les observations des chapitres précédents sont encore valables dans ces résultats. En plus de cela, on peut noter la faible variabilité de chacun des paramètres lorsque l'on passe des durées d'exécution déterministes aux durées d'exécution aléatoires et aux pannes aléatoires. Cette faible variabilité est confirmée par des tests statistiques sur la différence entre les paramètres moyens dans les cas notés DP, ceux dans les cas notés RP et ceux dans les cas notés RPF. Cette faible variabilité des résultats moyens traduit une grande robustesse des approches de prise de décision proposées dans des conditions d'utilisation réelles. Le lecteur peut se référer à [7] pour des analyses additionnelles sur certaines de ces simulation.

Chapitre 6

Conclusion générale et perspectives

6.1 Conclusions

Dans cette thèse, nous avons proposé des approches de solution à des problèmes de prise de décision en temps-réel dans les processus de maintenance des systèmes distribués avec des ressources de maintenance partagées. Face à l'évolution des systèmes technologiques avec des aspects de plus en plus répartis, le concept de systèmes distribués est de plus en plus d'actualité et en maintenance, les problèmes de gestion coopérative de la maintenance font de plus en plus surface. Le lecteur peut trouver des analyses sur les besoins de gestion coopérative de la maintenance des systèmes distribués dans [1].

Il est évident que la réussite d'un processus de maintenance en général, et de la maintenance coopérative des systèmes distribués en particulier, dépend de la réactivité avec laquelle les ressources de maintenance sont déployées dans l'accomplissement des tâches de maintenance. Une grande réactivité permet de maîtriser les coûts supportés et la réactivité dans les interventions de maintenance dépend des approches de prise de décision en temps-réel dans le processus de maintenance. En général, le responsable de maintenance doit faire face, avec un nombre très réduit de ressources de maintenance, à des urgences qui, plus qu'en production, ont un aspect particulier en maintenance. Dans cette situation, il doit souvent affronter des dilemmes et décider, entre deux maux, de résoudre l'un en premier lieu plutôt que l'autre. Les délais d'intervention des ressources de maintenance, l'efficacité de la politique de maintenance pratiquée et les coûts supportés, dépendent principalement des méthodes de prise de décision adoptées en temps-réel. Des approches adéquates doivent lui permettre de prendre ces décisions avec un moindre risque et un moindre coût. Ce travail n'a pas la prétention de remplacer le responsable de maintenance mais d'aller dans le sens de l'aide à la décision et de faciliter le travail de ce dernier.

Parmi les différents types de coûts abordés dans ce document, se situent des coûts dus notamment à des états de fonctionnement en dégradation avancée des systèmes, états que nous avons appelés dans tout ce document états critiques. Ces coûts se rapportent principalement à des temps de réponse (pondérés ou non) des tâches de maintenance. Il y a d'un autre côté des coûts qui se rapportent à des retards de maintenance, et aussi des coûts relatifs à la maintenance juste à temps,

en analogie avec le concept de *JIT* - "*Just In Time*" en production. Ces différents coûts peuvent être vus séparément ou être traités conjointement. C'est ainsi que nous avons travaillé avec des combinaisons de ces différents coûts.

Le traitement des ressources de maintenance pour l'ensemble des sites du système distribué impose d'importantes contraintes logistiques qui dépendent du déplacement de ces ressources d'un site à un autre. Les temps logistiques que cela implique sont dépendants de la séquence d'exécution des tâches de maintenance. Ces temps logistiques dépendants de la séquence des tâches de maintenance, rendent particulièrement délicate la prise de décision par rapport à chaque ressource de maintenance devant être déplacée d'un site à un autre. Les problèmes abordés sont NP-difficiles. Par ailleurs, dans le souci de proposer des approches permettant la prise de décision en temps-réel avec une grande réactivité face aux sollicitations des sites du système, les approches que nous avons proposées se basent sur des règles d'optimalité locale suivant le type de coût abordé et suivant la configuration du système. Ces règles d'optimalité locale, conjointement avec les bornes inférieures que nous avons proposées sur leur utilisation en statique, peuvent servir dans d'autres types de problèmes comme nous allons en l'expliquer dans la section suivante.

En outre, les domaines de gestion de production et de gestion de maintenance ont, jusqu'à présent évolué chacun de leur côté et n'ont pas vraiment eu d'inter-action dans la recherche de solutions aux problèmes qu'ils affrontent. Pourtant, certaines préoccupations ont des correspondances dans l'un et l'autre de ces domaines. Les méthodes servant dans l'un de ces domaines peuvent, sinon être directement adaptées, au moins avoir des extensions pour la résolution de certains problèmes dans l'autre. Ce travail se veut d'être un pont entre ces deux domaines. En effet, pour résoudre les problèmes de gestion temps-réel des décisions de maintenance, les approches de prise de décision que nous avons proposées se basent sur des rapprochements avec des problèmes en gestion de production. Nous avons, pour une ressource de maintenance, fait des rapprochements avec les problèmes d'ordonnancement sur une machine "*single machine scheduling problem*", pour résoudre les problèmes posés. Nous espérons que ceci soit le point de départ pour une longue co-habitation entre ces deux domaines.

Les approches proposées ont été illustrées sur des exemples simples puis utilisées sur des systèmes de taille réelle. Elles partent dans un premier temps des durées d'exécution déterministes relatives à chacune des entités dans le système. Elles utilisent la fonction disponibilité qui indique l'état de dégradation des entités du système. Elles ont ensuite été utilisées en environnement incertain avec des aléas (durées d'exécution aléatoires des tâches de maintenance, pannes fortuites) dans le fonctionnement du système. Cette utilisation en environnement incertain s'est faite à l'aide de la simulation Monté-Carlo. Les résultats obtenus montrent une grande robustesse des méthodes. Ces approches fournissent des outils d'aide à la décision pour la maintenance et peuvent être intégrées dans des logiciels de gestion de maintenance assistée par ordinateur .

6.2 Perspectives

Les problèmes abordés dans ce documents se situent dans une gamme très vaste de problèmes se rapportant à la maintenance coopérative des systèmes distribués. Ils ouvrent ainsi la voie à des problèmes voisins et complémentaires dont certains sont dans nos objectifs immédiats pour la suite de ce travail.

6.2.1 Problèmes avec des ressources de différentes classes

Dans les modèles proposés, nous avons considéré une seule classe de ressources de maintenance. Ces ressources sont supposées avoir les mêmes performances. En fait, lorsqu'il y a différentes classes de ressources de maintenance, chaque classe de ressources est consacrée à un aspect particulier de la maintenance. Ainsi, pour un système ayant plusieurs classes de ressources, une subdivision en sous-systèmes avec une classe de ressources pour chaque sous-système peut être faite. De cette façon, les problèmes dans chacun de ces sous-systèmes peuvent se ramener à des problèmes avec une classe de ressources et être traités avec les approches présentées dans ce document. L'un de nos objectifs dans la suite immédiate de ce travail est d'utiliser les approches proposées sur des systèmes avec différents sous-systèmes nécessitant différentes classes de ressources d'un sous-système à un autre.

6.2.2 Maintenance avec fenêtres temporelles imposées

Les problèmes de fenêtres temporelles sont des problèmes qui surviennent naturellement en gestion de la maintenance. En effet, dans un système de production soumis à des actions de maintenance, la production est souvent considérée prioritaire et, sur un horizon de planification, les actions de maintenance (autres que de réparation suite à des pannes), sont repoussées et regroupées sur des périodes très précises données, compte tenu des contraintes de production. Cela peut être par exemple un jour accordé pour regrouper toutes les interventions de maintenance préventive en attente au cours d'une semaine, au cours d'un mois ou quelques jours groupés accordés pour les actions en attente au cours d'un trimestre, d'un semestre. Sur une année, cela peut être quelques périodes, chacune de quelques jours.

Les différents problèmes qui peuvent se poser consistent notamment à :

- maximiser le nombre total d'interventions réalisées au cours de ces périodes imposées ;
- minimiser le nombre total des interventions ne pouvant être réalisées au cours de ces périodes accordées (ce qui est un problème dual du précédent) ;
- minimiser les coûts des états critiques des entités du système, sachant que les interventions de maintenance préventive ne sont exécutées que sur ces périodes imposées ;
- minimiser les coûts dus à des retards de maintenance sur le système, sachant que les interventions de maintenance préventive ne sont exécutées que sur ces périodes imposées ;
- minimiser les coûts relatifs à la maintenance juste à temps, sachant que les interventions de

maintenance préventive ne sont exécutées que sur ces périodes ;

- déterminer (dans le cas où l'amplitude des périodes est imposée, mais non leur positionnement sur l'horizon de planification) le positionnement de ces périodes afin de minimiser les différents coûts qui surviennent ;
- déterminer le positionnement et/ou la longueur (l'amplitude) de ces périodes afin de trouver le meilleur compromis entre la production et l'équipe de maintenance et, si possible le compromis optimal, etc.

Ces différents problèmes méritent qu'on leur apporte des solutions afin de permettre une bonne intégration des politiques de maintenance dans les politiques de production. Nous y reviendrons dans un futur proche.

6.2.3 Étude de caractéristiques asymptotiques dans les politiques proposées

Comme nous l'avons relevé dans les analyses de l'état de l'art, les problèmes de maintenance abordés dans la littérature concernent en grande partie l'étude de caractéristiques asymptotiques des systèmes sous des politiques de maintenance données. Un des objectifs qui va dans ce sens par rapport aux approches de décision porte justement sur cet aspect d'études de paramètres asymptotiques sur les entités du système. Ces études peuvent porter sur le taux de défaillance asymptotique de chacune des entités du système dans les politiques proposées, la disponibilité asymptotique de chacune des entités séparément, la disponibilité asymptotique de chacun des sites du système, la disponibilité asymptotique globale du système. En plus de ces paramètres, il y a la disponibilité opérationnelle de chacun des sites du système et la disponibilité opérationnelle globale du système et ce, aussi bien sur l'horizon de planification considéré que sur le plan asymptotique. D'autres paramètres asymptotiques du système peuvent être analysés en plus de ceux-ci.

6.2.4 Méthodes exactes avec les outils développés

Dans l'optique de proposer des approches de prise de décision en temps-réel dans le processus de maintenance, pour des problèmes qui sont NP-difficiles, nous n'avons pas particulièrement insisté sur la recherche de méthodes exactes de résolution. En raison de l'exigence de réactivité dans ce processus, nous avons plutôt mis l'accent sur des méthodes rapides de décision basées surtout sur des règles d'optimalité locale. En plus des règles utilisées qui existent déjà dans la littérature, ces règles concernent principalement des problèmes, avec dates d'occurrence inégales des tâches de maintenance, de la minimisation de :

- temps de réponse pondéré des tâches de maintenance en absence de temps logistiques entre les tâches,
- temps de réponse non pondéré des tâches de maintenance avec des temps logistiques dépendants de la séquence de ces tâches,
- temps de réponse pondéré des tâches de maintenance en présence de temps logistiques dépendants de la séquence des tâches,

- combinaison convexe ou linéaire du temps de réponse et du retard des tâches de maintenance (en l'absence et en présence de temps logistiques dépendants de la séquence des tâches),
- coût d'avance et retard des tâches de maintenance (en l'absence et en présence de temps logistiques dépendant de la séquence des tâches).

Ces règles sont établies sur des problèmes statiques ou localement statiques, et nous avons proposé des bornes inférieures dans le sens de ces problèmes statiques, avant leur utilisation pour la décision en temps-réel. Avec ces règles définissant des propriétés de dominance et les bornes inférieures, les conditions sont réunies pour le développement de méthodes exactes de résolution de ces problèmes statiques, du type *branch and bound*. L'un de nos objectifs dans le futur immédiat est de travailler sur ces méthodes exactes.

Appendice A

Concepts de base

A.1 Introduction

Ce chapitre est consacré à la définition et aux propriétés des concepts principaux utilisés dans ce document. Il s'agit principalement des notions de maintenabilité, maintenance, disponibilité, défaillance et réparation. Ces différentes définitions se basent sur les normes françaises en vigueur et les propriétés qui en découlent sont des conséquences de leurs définitions probabilistes.

Ces termes se regroupent au sein de la sûreté de fonctionnement. C'est un domaine qui a largement été développé dans la deuxième moitié du vingtième siècle. Avec les développements dont il a été l'objet et ses différentes applications, il est devenu une science à part entière. Il convient ainsi de voir comment cette science est définie.

A.2 Définitions

A.2.1 Système

Dans tout ce document, le mot *système* est omniprésent. La définition que nous en donnons est une définition qui ressort de [163] et [164].

Un *système* est un ensemble déterminé d'éléments discrets (ou composants) interconnectés ou en interaction. Cette définition implique que le système est fait de composants en interaction. On suppose ainsi que le système n'est pas simplement la somme de ses sous-systèmes ou de ses composants. En outre, si la nature physique d'un sous-système ou d'un composant se modifie suite à une défaillance, le système est lui-même modifié [163], [164].

A.2.2 Défaillance

La *défaillance* est définie comme la "*cessation de l'aptitude d'une entité à accomplir une fonction requise*". On dira qu'une entité connaît une défaillance lorsqu'elle n'est plus en mesure de remplir sa (ses) fonction(s). Par extension, on considère parfois qu'il y a une défaillance lorsqu'il y a

altération de l'aptitude d'une entité à accomplir une fonction requise. Les tolérances associées doivent alors être définies. [163], [164] [29], [32], [31], [33], [65], [66], [81], [111], [174]

On considère un ensemble d'entités réputées identiques et en fonctionnement depuis un instant initial. On définit le *taux de défaillance* de cette entité comme la proportion, ramenée à l'unité de temps, des entités qui, ayant survécu à un instant t arbitraire, ne sont plus en vie à l'instant $t + \Delta t$; par passage à la limite, on obtient un *taux de défaillance instantané* qui est fonction de t . (Nous en reparlerons).

A.2.3 Panne

La *panne* est l' "*inaptitude d'une entité à accomplir une fonction requise*". Après l'apparition d'une défaillance, on considère que l'entité est en panne ; une panne résulte toujours d'une défaillance [163], [164], [29], [32], [31], [33], [65], [66], [81], [111]. On peut théoriquement imaginer l'existence d'une panne sans qu'il y ait eu cessation de l'aptitude à accomplir une fonction requise - et donc défaillance - si l'entité n'a jamais été apte à fonctionner. Ceci pourrait être dû à une erreur de conception.

La figure A.1 dresse les relations entre ces différentes notions tel que l'a établi Villemeur [163], [164].

A.3 Sûreté de fonctionnement

Au sens large, la *sûreté de fonctionnement* est définie comme la *science des défaillances* ; elle inclut ainsi leur connaissance, leur évaluation, leur prévision, leur mesure et leur maîtrise [163], [164], [29], [96], [101]. Au sens strict, la sûreté de fonctionnement est l'aptitude d'une entité à satisfaire à une ou plusieurs fonctions requises dans des conditions données. Elle peut être caractérisée par les concepts de fiabilité, de disponibilité, de maintenabilité, de sécurité.

La disponibilité et la maintenabilité étant les concepts qui nous intéressent le plus dans ce document, nous les développerons davantage par la suite. Néanmoins, il convient de définir au moins succinctement la fiabilité et la sécurité.

A.3.1 Fiabilité

La *fiabilité* d'une entité E est définie par son "*aptitude à accomplir une fonction requise dans des conditions données pendant un intervalle de temps donné*" [Norme NF X 60-500]. En terme probabiliste, la fiabilité d'une entité E est définie comme suit :

$$R(t) = \mathbb{P}(E \text{ non défaillant sur } [0,t])$$

L'aptitude contraire de la fiabilité est la *défiabilité* dont l'expression probabiliste est la suivante à l'instant t :

$$\bar{R}(t) = 1 - R(t)$$

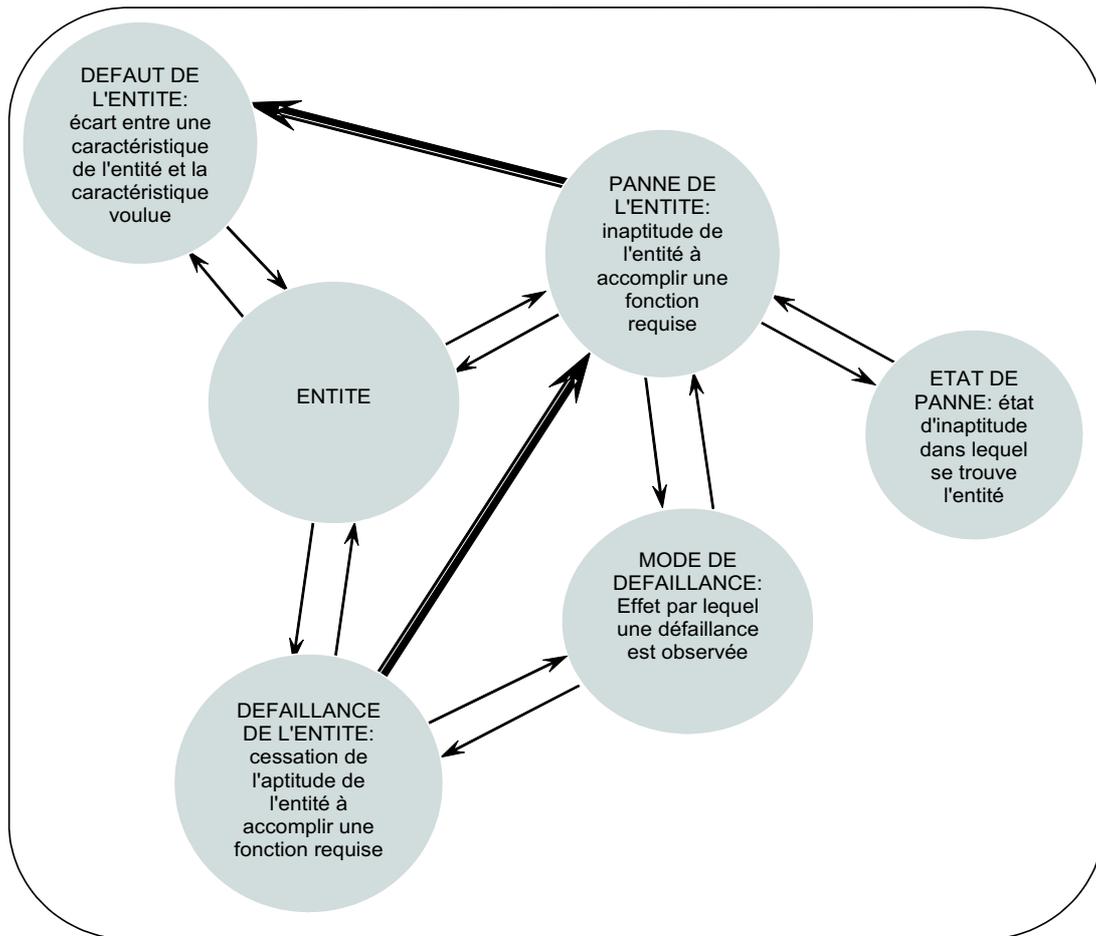


Figure A.1 – Défaut, défaillance et panne d'un système

A.3.2 Sécurité

On définit la *sécurité* d'une entité E comme son "*aptitude à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques*" [163], [164], [29], [96], [101]. L'aptitude contraire de la sécurité est l'*insécurité*. Tout comme la fiabilité, la sécurité se définit par une probabilité.

A.4 Disponibilité

La *disponibilité* d'une entité E est définie comme son "*aptitude à être en état d'accomplir une fonction requise dans des conditions données, à un instant donné ou pendant un intervalle de temps donné, en supposant que la fourniture de moyens extérieurs nécessaire soit assurée*" [Norme X 60-500]. Elle est en général définie en terme probabiliste comme suit à un instant t :

$$A(t) = \mathbb{P}(E \text{ non défaillant à l'instant } t)$$

L'aptitude contraire de la disponibilité est l'*indisponibilité* définie par :

$$\bar{A}(t) = 1 - A(t)$$

Il faut noter que la disponibilité d'une entité est en général une fonction non croissante du temps t et donc, a fortiori, l'indisponibilité est une fonction non décroissante du temps t .

A.4.1 Disponibilité de deux composants en série

Pour un système S formé de deux entités E_1 et E_2 en série fonctionnant de façon indépendante, la disponibilité à l'instant t est donnée par

$$A_S(t) = A_{E_1}(t) \cdot A_{E_2}(t)$$

En effet, étant donné que les fonctionnements de E_1 et E_2 , sont indépendants, on a :

$$\begin{aligned} A_S(t) &= \mathbb{P}(S \text{ non défaillant à l'instant } t) \\ &= \mathbb{P}(E_1 \text{ non défaillant et } E_2 \text{ non défaillant à l'instant } t) \\ &= \mathbb{P}(E_1 \text{ non défaillant à l'instant } t) \cdot \mathbb{P}(E_2 \text{ non défaillant à l'instant } t) \\ &= A_{E_1}(t) \cdot A_{E_2}(t) \end{aligned}$$

A.4.2 Disponibilité de deux composants en parallèle

Si un système S est formé de deux entités E_1 et E_2 en parallèle fonctionnant de façon indépendante, sa disponibilité à l'instant t est donnée par :

$$A_S(t) = 1 - (1 - A_{E_1}(t))(1 - A_{E_2}(t))$$

En effet, avec le fonctionnement indépendant de E_1 et E_2 , l'indisponibilité de S à l'instant t est :

$$\begin{aligned} 1 - A_S(t) &= \mathbb{P}(S \text{ indisponible à l'instant } t) \\ &= \mathbb{P}(E_1 \text{ indisponible et } E_2 \text{ indisponible à l'instant } t) \\ &= \mathbb{P}(E_1 \text{ indisponible à l'instant } t) \cdot \mathbb{P}(E_2 \text{ indisponible à l'instant } t) \\ &= (1 - A_{E_1}(t))(1 - A_{E_2}(t)) \end{aligned}$$

d'où la disponibilité de S à l'instant t est :

$$A_S(t) = 1 - (1 - A_{E_1}(t))(1 - A_{E_2}(t))$$

A partir de ces expressions simples de la disponibilité de deux composants en série et en parallèle, on pourra exprimer la disponibilité d'un système quelconque dans lequel il y a aussi bien des entités en série que des entités en parallèle à partir des disponibilités individuelles de chacun des composants du système.

A.4.3 Expression de la disponibilité

L'expression générale de la disponibilité d'une entité dépend de son taux de défaillance et de son taux de réparation. (Nous définirons plus clairement ces deux concepts après). Nous nous limiterons ici à un composant dont le taux de défaillance (λ) et le taux de réparation (μ) sont constants.

La disponibilité à l'instant t d'une entité dont le taux de défaillance et le taux de réparation sont

constants est donnée par:

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} \exp(-(\lambda + \mu)t)$$

Et par conséquent, l'indisponibilité est donnée par:

$$\bar{A}(t) = \frac{\lambda}{\lambda + \mu} (1 - \exp(-(\lambda + \mu)t))$$

Ainsi, la disponibilité asymptotique et l'indisponibilité asymptotique, à l'infini sont respectivement définies par

$$A(\infty) = \frac{\mu}{\lambda + \mu} \text{ et } \bar{A}(\infty) = \frac{\lambda}{\lambda + \mu}$$

Nous les définirons par la suite en fonction du *MTTR* et du *MTTF*.

Les grandeurs moyennes associées à la disponibilité le plus souvent utilisées sont :

- le Temps Moyen de Disponibilité (MUT - Mean Up Time)
- le Temps Moyen d'indisponibilité (MDT - Mean Down Time).

A.5 Maintenabilité et maintenance

Dans des conditions données d'utilisation, la *maintenabilité* d'une entité est son "*aptitude à être maintenue ou rétablie, sur un intervalle de temps donné, dans un état dans lequel elle peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et moyens prescrits*" [Norme NF X 60-500].

La *maintenance* elle, est définie comme l' "*ensemble des actions destinées à rétablir une entité dans un état dans lequel elle peut accomplir une fonction requise*" [Norme NF X 60-500].

En général, la maintenabilité se caractérise par une probabilité $M(t)$ (également appelée maintenabilité) que la maintenance d'une entité E soit accomplie dans des conditions données, avec des procédures et moyens prescrits, soit achevée au temps t , sachant que E est défaillant à l'instant initial.

Remarquons que la maintenabilité est en général une fonction non décroissante du temps t . Elle varie de 0 à 1 sur $[0; +\infty[$ et a une limite égale à 1 à plus l'infini.

$$\begin{aligned} M(t) &= \mathbb{P}(\text{la maintenance de } E \text{ est achevée au temps } t) \\ &= 1 - \mathbb{P}(E \text{ non réparée sur la durée } [0, t]) \end{aligned}$$

L'aptitude contraire de la maintenabilité est l'*immaintenabilité*. Elle est mesurée par :

$$\bar{M}(t) = 1 - M(t)$$

La maintenabilité et l'immaintenabilité sont des concepts qui ne concernent que les entités et systèmes réparables. Différents types de maintenance sont distingués et peuvent être regroupée comme suit :

- La maintenance préventive qui a pour but de réduire la probabilité de défaillance d'un système ; il y a en particulier la maintenance préventive systématique planifiée et la maintenance

conditionnelle qui est entreprise lorsque des comportements non souhaités sont décelés dans le fonctionnement du système. C'est de la maintenance préventive subordonnée à un type d'événement prédéterminé (auto diagnostic, information d'un capteur, mesures d'usures, etc).

- La maintenance corrective qui intervient après défaillance.

A.6 Taux de défaillance, de réparation et de remise en service

Ces grandeurs sont celles qui caractérisent essentiellement la défaillance et la maintenabilité (et donc la maintenance) d'un système.

A.6.1 Taux de défaillance

Le *taux de défaillance (instantané)* $\lambda(t)$ d'une entité E est la "limite, si elle existe, du quotient de la probabilité conditionnelle pour que l'instant t d'une défaillance de l'entité soit compris entre $[t, t + \Delta t]$, par la durée Δt de l'intervalle de temps, lorsque Δt tend vers zéro en supposant que l'entité n'a pas eu de défaillance sur $[0, t]$ " [Norme NF X 60-500].

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \left(\frac{1}{\Delta t} \mathbb{P}(E \text{ défailant sur } [t, t + \Delta t] \text{ sachant qu'elle n'a pas eu de défaillance sur } [0, t]) \right)$$

Le taux de défaillance peut s'exprimer en fonction de la fiabilité $R(t)$. Cela se fait en utilisant le théorème des probabilités conditionnelles

$$\begin{aligned} \lambda(t) &= \lim_{\Delta t \rightarrow 0} \left(\frac{1}{\Delta t} \frac{R(t) - R(t + \Delta t)}{R(t)} \right) \\ &= -\frac{1}{R(t)} \frac{d}{dt} R(t) \\ &= \frac{1}{R(t)} \frac{d}{dt} \bar{R}(t) \end{aligned}$$

La fonction définie par $\frac{d}{dt} \bar{R}(t)$ est la densité de défaillance ou de défiabilité.

A.6.2 Taux de réparation

Le *taux de réparation (instantané)* $\mu(t)$ d'une entité E est la "limite, si elle existe, du quotient de la probabilité conditionnelle pour que l'instant t d'achèvement de la réparation de l'entité soit compris dans un intervalle de temps $[t, t + \Delta t]$, par la durée Δt de l'intervalle de temps, lorsque Δt tend vers zéro, en supposant que l'entité a été en panne sur $[0, t]$ " [Norme NF X 60-500].

$$\mu(t) = \lim_{\Delta t \rightarrow 0} \left(\frac{1}{\Delta t} \mathbb{P}(E \text{ est remise en service entre } t \text{ et } t + \Delta t \text{ sachant qu'elle a été en panne sur } [0, t]) \right)$$

En utilisant le théorème des probabilités conditionnelles, on peut aussi écrire le taux de réparation en fonction de la maintenabilité.

$$\begin{aligned}\mu(t) &= \lim_{\Delta t \rightarrow 0} \left(\frac{1}{\Delta t} \frac{M(t + \Delta t) - M(t)}{1 - M(t)} \right) \\ &= \frac{1}{1 - M(t)} \frac{d}{dt} M(t)\end{aligned}$$

A partir de ces relations, si on suppose que l'on a une entité de taux de défaillance et de réparation constants, alors en se basant sur le fait que la maintenabilité à l'instant initial est nulle et que la fiabilité à l'instant initiale est égale à 1, on a les expressions suivantes pour la maintenabilité et la fiabilité à un instant t

$$M(t) = 1 - \exp(-\mu t) \text{ et } R(t) = \exp(-\lambda t)$$

A.6.3 Taux de remise en service

Le *taux de remise en service (instantané)* $v(t)$ est défini comme la "limite, si elle existe, du quotient de la probabilité conditionnelle pour qu'une entité soit remise en service dans un intervalle de temps $[t, t + \Delta t]$, sachant qu'elle a une défaillance à l'instant 0 et que la remise en service n'a pas encore été effectuée à l'instant t , par la durée de l'intervalle de temps Δt , lorsque Δt tend vers zéro" [Norme NF X 60-500].

Il faut remarquer qu'en général, les fins de réparation et les remises en service ne sont pas les mêmes. Mais souvent, les deux terminologies sont confondues et l'on utilise le taux de remise en service comme taux de réparation.

A.7 Autres définitions

D'autres termes sont couramment utilisés en sûreté de fonctionnement. Nous en définirons certains qui sont les plus usuels dans cette section.

A.7.1 Le MTTF (Mean Time To Failure)

Le *MTTF* est la durée moyenne d'une entité avant la première défaillance. Elle est caractérisée par la défiabilité. C'est la moyenne (espérance mathématique) du temps de défiabilité. Il est aussi égal à l'opposé de la moyenne (espérance mathématique) du temps de fiabilité.

$$\begin{aligned}MTTF &= \int_0^{+\infty} t \cdot \frac{d}{dt} \bar{R}(t) dt \\ &= - \int_0^{+\infty} t \cdot \frac{d}{dt} R(t) dt\end{aligned}$$

Pour une entité dont le taux de défaillance est constant, l'expression précédente de la fiabilité nous permet de voir que $MTTF = \frac{1}{\lambda}$.

A.7.2 Le MTTR (Mean Time to Repair) ou (Mean Time To Restoration)

Le *MTTR* est la durée moyenne de réparation d'une entité. Il est caractérisé par la maintenabilité. Le *MTTR* est la moyenne (espérance mathématique) du temps de maintenabilité.

$$MTTR = \int_0^{+\infty} t \cdot \frac{d}{dt} M(t) dt$$

Dans le cas où $\lim_{t \rightarrow +\infty} [t(1 - M(t))] = 0$, cette moyenne peut se ramener, avec une intégration par partie, à :

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

Pour une entité réparable de taux de réparation constante, on retrouve avec l'expression précédente de la maintenabilité que $MTTR = \frac{1}{\mu}$.

A.7.3 Et aussi...

- Le MUT (Mean Up Time) : c'est la durée moyenne de fonctionnement après réparation.
- Le MDT (Mean Down Time) qui est la durée moyenne d'indisponibilité. Elle comprend :
 - la détection de la panne ;
 - la réparation de la panne ;
 - la remise en service.

Le MDT est la moyenne des durées d'indisponibilité suite à des défaillances.

- Le MTBF (Mean Time Between Failure) : C'est la durée moyenne entre deux défaillances consécutives d'une entité réparée.

Tous ces concepts se résument sur la figure qui suit .

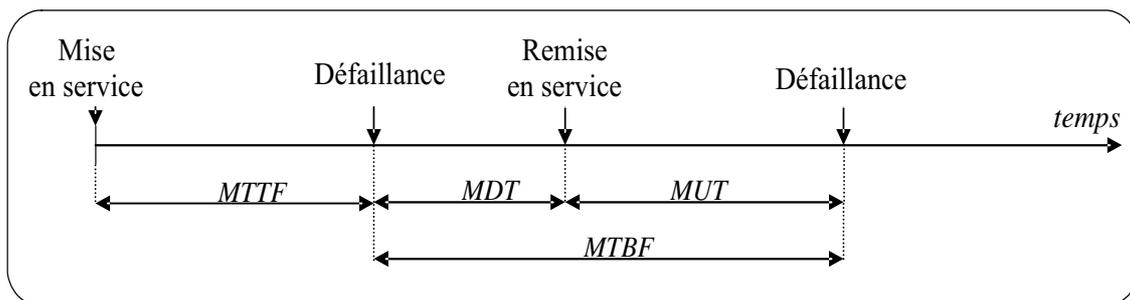


Figure A.2 – Grandeurs moyennes de sûreté de fonctionnement

Remarque A.7.3.1. Il ressort de la sous-section A.4.3 et de la présente section que la disponibilité et l'indisponibilité asymptotiques peuvent s'écrire en fonction du *MTTF* et du *MTTR* pour une entité dont le taux de défaillance et de réparation sont constants.

La disponibilité asymptotique et l'indisponibilité asymptotique s'expriment respectivement par

$$A(\infty) = \frac{MTTF}{MTTF + MTTR} \text{ et } \bar{A}(\infty) = \frac{MTTR}{MTTF + MTTR}$$

Appendice B

La disponibilité dans les cas non exponentiels

B.1 Introduction

La disponibilité des composants est une fonction très utilisée en organisation de la maintenance. Cette fonction permet de situer l'état d'un composant dans le temps et de prendre les décisions qui s'y rapportent. Cependant, son utilisation dans la littérature se réduit très souvent à la fonction disponibilité avec une fiabilité qui suit une loi exponentielle. La raison en est la simplicité de manipulation qu'elle procure mais aussi que, dès lors que la loi de fiabilité sort de ces cadres exponentiels, la disponibilité devient une fonction assez "complexe". La large utilisation faite de la fonction disponibilité dans le cas de la loi exponentielle peut être observée non seulement dans les travaux de la littérature mais aussi dans les livres consacrés à la maintenance dont notamment [17], [32], [93] et les autres livres référencés dans l'appendice A.

Dans ce chapitre, nous tenterons d'analyser le comportement de la fonction disponibilité dans des cas où la loi de disponibilité n'est pas exponentielle. Après avoir abordé le cas très général où les taux de défaillance et de réparation du composant sont très généraux mais analytiques, nous verrons comment le cas de la loi exponentielle se place dans ce cas général. Nous considérerons ensuite la fonction disponibilité lorsque la fiabilité suit une loi de Weibull et que le taux de réparation du composant est analytique. Nous partirons de ce cas général pour étudier la fonction disponibilité lorsque la fiabilité suit une loi de Weibull et que le taux de réparation est constant.

B.2 La fonction disponibilité dans le cas général

Dans cette section, nous étudions la fonction disponibilité dans le cas général. Nous supposons pour cela que les fonctions caractérisant les taux de défaillance et de réparation des entités ont des expressions analytiques. Même si ces expressions ne sont pas connues, les résultats tiennent compte de leur forme générale. Ainsi, la sous-section suivante donne l'équation différentielle que

suit la fonction disponibilité d'une entité dans le cas général.

B.2.1 Équation de la fonction disponibilité dans le cas général

L'équation différentielle que suit la disponibilité dans le cas général est donnée par le théorème suivant.

Théorème B.2.1.1. Équation différentielle générale de la disponibilité

Soit une entité E ayant un taux de défaillance $\lambda(t)$ et un taux de réparation $\mu(t)$. Supposons que ces caractéristiques puissent s'exprimer de façon analytique en fonction du temps.

Alors, la fonction disponibilité A de cette entité, qui, à t , associe $A(t)$ de cette entité vérifie l'équation différentielle ci-après :

$$\frac{d}{dt}A(t) + (\mu(t) + \lambda(t))A(t) = \mu(t) \quad (\text{B.2.1})$$

Preuve. Soit $A : t \mapsto A(t)$ la fonction disponibilité de l'entité. Soit dt un infiniment petit du temps. D'après la définition de la fonction disponibilité, nous avons :

$$\begin{aligned} A(t + dt) &= \mathbb{P}(E \text{ en fonction à l'instant } t \text{ et } E \text{ non défaillant sur la période } [t, t + dt] \\ &\quad \text{ou } E \text{ défaillant à l'instant } t \text{ et réparé sur la période } [t, t + dt]) \\ &= \mathbb{P}(E \text{ en fonction à l'instant } t \text{ et } E \text{ non défaillant sur la période } [t, t + dt]) \\ &\quad + \mathbb{P}(E \text{ défaillant à l'instant } t \text{ et réparé sur la période } [t, t + dt]) \\ &= \mathbb{P}(E \text{ en fonction à l'instant } t) \cdot \mathbb{P}(E \text{ non défaillant sur la période } [t, t + dt]) \\ &\quad + \mathbb{P}(E \text{ défaillant à l'instant } t) \cdot \mathbb{P}(E \text{ réparé sur la période } [t, t + dt]) \\ &= A(t) \cdot (1 - \lambda(t)dt) + (1 - A(t)) \cdot \mu(t)dt \end{aligned} \quad (\text{B.2.2})$$

Cette égalité donne l'équation suivante

$$A(t + dt) - A(t) = [\mu(t) - (\mu(t) + \lambda(t)) \cdot A(t)]dt \quad (\text{B.2.3})$$

ou encore

$$\frac{A(t + dt) - A(t)}{dt} = \mu(t) - (\mu(t) + \lambda(t)) \cdot A(t) \quad (\text{B.2.4})$$

Cette dernière équation B.2.4 donne, lorsque dt tend vers 0

$$\frac{d}{dt}A(t) + (\mu(t) + \lambda(t)) \cdot A(t) = \mu(t) \quad (\text{B.2.5})$$

□

Ce théorème permet d'en déduire l'expression générale de la fonction disponibilité.

B.2.2 Expression de la fonction disponibilité dans le cas général

A partir du théorème B.2.1.1, l'expression de la fonction disponibilité dans le cas général est donnée dans le théorème suivant.

Théorème B.2.2.1. Expression de la fonction disponibilité dans le cas général

Soit une entité E ayant un taux de défaillance $\lambda(t)$ et un taux de réparation $\mu(t)$. Supposons que ces caractéristiques puissent s'exprimer en fonction du temps et que le taux de défaillance $\mu : t \mapsto \mu(t)$ et le taux de réparation $\lambda : t \mapsto \lambda(t)$ soit intégrables sur toute partie bornée de \mathbb{R} . Soit en plus a un réel tel que la fonction disponibilité prenne la valeur 1 en a .

Alors la fonction disponibilité A de cette entité, en tout réel $t \geq a$ est donnée par l'expression suivante :

$$A(t) = \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \cdot \left\{1 + \int_a^t \mu(x) \cdot \exp\left(\int_a^x [\mu(u) + \lambda(u)]du\right) dx\right\} \quad (\text{B.2.6})$$

Ce théorème donne l'expression de la fonction disponibilité dans le cas général. A partir de là, la fonction disponibilité pourra être étudiée dans tous les cas où le taux de défaillance λ et le taux de réparation μ peuvent être exprimés en fonction du temps t .

Preuve. Pour résoudre l'équation différentielle de la fonction disponibilité donnée dans l'expression B.2.1 et que nous rappelons ci-après, nous allons commencer par l'équation homogène.

$$\frac{d}{dt}A(t) + (\mu(t) + \lambda(t)).A(t) = \mu(t)$$

L'équation homogène de cette équation différentielle est la suivante :

$$\frac{d}{dt}A(t) + (\mu(t) + \lambda(t)).A(t) = 0 \quad (\text{B.2.7})$$

Cette équation B.2.7 revient à

$$\frac{dA(t)}{A(t)} = -(\mu(t) + \lambda(t))dt \quad (\text{B.2.8})$$

ce qui donne, en résolvant cette équation B.2.8

$$A(t) = K \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \quad (\text{B.2.9})$$

où K est une constante réelle et a un autre réel fixé.

Maintenant, pour résoudre l'équation générale, nous allons procéder par une variation de la constante K . Supposons donc que K soit une fonction du temps t notée $K(t)$. Alors l'égalité B.2.9 devient

$$A(t) = K(t) \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \quad (\text{B.2.10})$$

et ainsi la dérivée de la fonction A s'exprime par :

$$\begin{aligned} \frac{d}{dt}A(t) &= \frac{d}{dt} \left\{ K(t) \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \right\} \\ &= \left\{ \frac{d}{dt}K(t) \right\} \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) - \\ &\quad K(t) \cdot (\mu(t) + \lambda(t)) \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \end{aligned} \quad (\text{B.2.11})$$

Cela donne donc

$$\begin{aligned} \frac{d}{dt}A(t) + [\mu(t) + \lambda(t)].A(t) &= \left\{ \frac{d}{dt}K(t) - K(t).[\mu(t) + \lambda(t)] + K(t).[\mu(t) + \lambda(t)] \right\} \\ &\quad \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \\ &= \left\{ \frac{d}{dt}K(t) \right\} \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \end{aligned} \quad (\text{B.2.12})$$

Cette équation B.2.12, ensemble avec l'équation B.2.6 impliquent que

$$\left\{ \frac{d}{dt}K(t) \right\} \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) = \mu(t) \quad (\text{B.2.13})$$

soit

$$\frac{d}{dt}K(t) = \mu(t) \cdot \exp\left(\int_a^t [\mu(x) + \lambda(x)]dx\right) \quad (\text{B.2.14})$$

d'où

$$K(t) = \int_a^t \left(\mu(x) \cdot \exp\left(\int_a^t [\mu(u) + \lambda(u)]du\right) \right) dx + C \quad (\text{B.2.15})$$

où C est une constante réelle.

Ainsi, de l'équation B.2.10 on obtient

$$A(t) = \left\{ \int_a^t \left(\mu(x) \cdot \exp\left(\int_a^t [\mu(u) + \lambda(u)]du\right) \right) dx + C \right\} \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \quad (\text{B.2.16})$$

Finalement, si le réel a est tel que $A(a) = 1$, alors nous avons $C = 1$, et ainsi

$$A(t) = \left\{ \int_a^t \left(\mu(x) \cdot \exp\left(\int_a^t [\mu(u) + \lambda(u)]du\right) \right) dx + 1 \right\} \cdot \exp\left(-\int_a^t [\mu(x) + \lambda(x)]dx\right) \quad (\text{B.2.17})$$

□

Des résultats de cette section qui indiquent ce qui se passe dans le cas général sur la fonction disponibilité d'une entité, nous allons voir ce qui se passe dans le cas des certaines lois les plus usuelles en théorie de la fiabilité dont certaines, telle que la loi de Weibull, regroupent une gamme très large d'autres lois.

B.3 Disponibilité dans le cas d'une loi exponentielle

Dans cette section, nous allons montrer que la fonction disponibilité, dans le cas de la loi exponentielle, et qui est très largement utilisée, n'est qu'un cas particulier du cas général étudiée dans ce chapitre. Cette loi caractérise en général les entités électroniques, ou les entités dont le taux de défaillance λ et de réparation μ sont constants sur une période très grande de leur durée de vie.

B.3.1 Équation de la disponibilité dans le cas exponentiel

L'équation différentielle qui caractérise la fonction disponibilité dans le cas de la loi exponentielle peut être établie directement, ou bien déduite de l'équation différentielle dans le théorème B.2.1.1.

Proposition B.3.1.1. Équation différentielle de la disponibilité dans le cas exponentiel

Soit une entité E ayant un taux de défaillance λ et un taux de réparation μ constants.

Alors, la fonction disponibilité A qui à t associe $A(t)$ de cette entité vérifie l'équation différentielle ci-après :

$$\frac{d}{dt}A(t) + (\mu + \lambda)A(t) = \mu \quad (\text{B.3.1})$$

Preuve. Le taux de défaillance constant et le taux de réparation constant sont bien exprimés en fonction du temps et remplissent ainsi les hypothèses du théorème B.2.1.1. Cette équation est donc l'équation B.2.1 pour ce cas particulier. \square

B.3.2 Expression de la disponibilité dans le cas exponentiel

Si l'on ne savait pas ce qui se passe dans le cas général, l'expression de la disponibilité dans le cas de la loi exponentielle se retrouve en résolvant l'équation différentielle de la proposition B.3.1.1 précédent. Mais ici, cette expression se retrouve à partir de l'expression générale dans le théorème B.2.2.1.

Proposition B.3.2.1. Expression de la fonction disponibilité dans le cas de la loi exponentielle

Soit une entité E ayant un taux de défaillance λ et un taux de réparation μ constant.

Alors la fonction disponibilité A en tout réel $t \geq 0$ est donnée par l'expression suivante :

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} \exp(-[\lambda + \mu]t) \quad (\text{B.3.2})$$

Preuve. Le taux de défaillance λ constant et le taux de réparation μ constant sont bien exprimés en fonction du temps et ils représentent des fonctions intégrables sur toute partie bornée de \mathbb{R} . Les hypothèses du théorème B.2.2.1 sont donc bien vérifiées. Et le réel a vérifiant la condition $A(a) = 1$ est 0. Ainsi, à partir de l'expression B.2.6 nous avons pour tout réel $t \geq 0$,

$$\begin{aligned} A(t) &= \exp\left(-\int_0^t [\mu + \lambda]dx\right) \cdot \left\{1 + \int_0^t \mu \cdot \exp\left(\int_0^x [\mu + \lambda]du\right) dx\right\} \\ &= \exp(-[\mu + \lambda]t) \cdot \left\{1 + \int_0^t \mu \cdot \exp([\mu + \lambda]x) dx\right\} \\ &= \exp(-[\mu + \lambda]t) \cdot \left\{1 + \frac{\mu}{\mu + \lambda} \cdot [\exp((\mu + \lambda)x)]_0^t\right\} \\ &= \exp(-[\mu + \lambda]t) \cdot \left\{1 + \frac{\mu}{\mu + \lambda} \cdot \exp((\mu + \lambda)t) - \frac{\mu}{\mu + \lambda}\right\} \\ &= \exp(-[\mu + \lambda]t) \cdot \left\{\frac{\lambda}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} \cdot \exp((\mu + \lambda)t)\right\} \\ &= \frac{\lambda}{\mu + \lambda} \cdot \exp(-[\mu + \lambda]t) + \frac{\mu}{\mu + \lambda} \end{aligned} \quad (\text{B.3.3})$$

\square

Avant d'aborder l'étude de la disponibilité d'une entité dont la fiabilité suit la loi de Weibull, nous allons tout d'abord voir quelques caractéristiques de cette loi.

B.4 Caractéristiques de la loi de Weibull

B.4.1 Rappels et définitions

La loi de Weibull est la loi qui décrit en général le comportement de la fiabilité d'une entité mécanique. Elle est caractérisée de façon général par trois paramètres [59] qui sont:

1. Le paramètre de décalage : noté γ , ce paramètre représente une origine temporelle pour cette loi (dans le cas d'une variable aléatoire temporelle). Il prend ses valeurs dans \mathbb{R} tout entier. Lorsque γ est négatif, l'entité peut être défaillante à l'instant initial $t = 0$.
2. Le paramètre d'échelle : Il est noté σ et prend ses valeurs dans \mathbb{R}_+^* . Lorsque la loi de Weibull se ramène à une loi de exponentielle, σ est le temps moyen entre défaillances (MTBF) de l'entité.
3. Le paramètre de forme : Ce paramètre est noté β et prend ses valeurs dans \mathbb{R}_+^* . Il impose la forme de la loi et, comme nous le verrons plus loin, influe fortement sur la cinétique de dégradation de l'entité. Plus ce paramètre est grand, plus la dégradation est rapide. Les cas où $0 < \beta < 1$ correspondent à la période de jeunesse de l'entité. Les cas où $\beta = 1$ correspondent à la période de maturité de l'entité. Cette période peut être plus ou moins longue suivant le type d'entité ; elle détermine la forme aplatie ou recourbée de la courbe en baignoire représentant les variations du taux de défaillance. Les cas où $\beta > 1$ correspondent à la période de vieillesse de l'entité.

En théorie de fiabilité, la loi de Weibull paramétrée par γ , σ et β conduit à modéliser :

1. Le taux de défaillance par :

$$\lambda(t) = \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \text{ pour } t \geq \max(0, \gamma) \quad (\text{B.4.1})$$

2. La densité de probabilité de défaillance par :

$$-\frac{dR(t)}{dt} = \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \exp\left(-\left(\frac{t-\gamma}{\sigma}\right)^\beta\right) \text{ pour } t \geq \max(0, \gamma) \quad (\text{B.4.2})$$

3. La fiabilité par:

$$R(t) = \exp\left(-\left(\frac{t-\gamma}{\sigma}\right)^\beta\right) \text{ pour } t \geq \max(0, \gamma) \quad (\text{B.4.3})$$

Afin de mieux cerner le comportement de la fonction disponibilité dont l'étude se fera dans la suite, voyons comment évolue le taux de défaillance $\lambda(t)$ d'une entité soumise à une loi de Weibull.

B.4.2 Étude du taux de défaillance $\lambda(t)$ suivant la loi de Weibull

Comme il est rappelé plus haut, le taux de défaillance $\lambda(t)$ est défini par:

$$\begin{aligned} \lambda : [\max(0, \gamma), +\infty[&\longrightarrow \mathbb{R}_+ \\ t &\longmapsto \lambda(t) = \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \end{aligned} \quad (\text{B.4.4})$$

avec $\sigma \in \mathbb{R}_+^*$, $\beta \in \mathbb{R}_+^*$ et $\gamma \in \mathbb{R}$.

Les propriétés de ce taux de défaillance sont données dans la proposition suivante.

Proposition B.4.2.1. Variation du taux de défaillance suivant les valeurs du paramètre de forme

a) Si $0 < \beta < 1$, alors le taux de défaillance λ est une fonction strictement décroissante, hyperbolique, passant de $+\infty$ lorsque le temps $t = \gamma$, à 0 lorsque le temps t tend vers plus l'infinie (fig. B.1).

b) Si $\beta = 1$ alors le taux de défaillance λ est une fonction constante et vaut $\frac{1}{\sigma}$. La loi de Weibull devient alors une loi exponentielle (fig. B.1).

c) Si $1 < \beta < 2$, alors le taux de défaillance λ est une fonction strictement croissante, logarithmique (avec un axe parabolique dont la direction est l'axe du temps t). Cela signifie que λ a une croissance lente, et passe de 0 lorsque le temps $t = \gamma$ à plus l'infinie lorsque le temps t tend vers plus l'infinie (fig. B.1).

d) Si $\beta = 2$, alors le taux de défaillance λ est une fonction linéaire, strictement croissante, avec une pente de $\frac{2}{\sigma^2}$. Il passe de 0 lorsque le temps $t = \gamma$ à plus l'infinie lorsque le temps t tend vers plus l'infinie (fig. B.1).

e) Si $\beta > 2$, alors le taux de défaillance λ est une fonction strictement croissante exponentielle (avec un axe parabolique dont la direction est l'axe des λ). Il a donc une croissance rapide et passe de 0 lorsque le temps $t = \gamma$ à plus l'infinie lorsque le temps t tend vers plus l'infinie (fig. B.1).

Dans la démonstration de cette proposition, nous n'allons pas étudier les différents cas un à un, mais ces différents cas vont se distinguer dans le fil du raisonnement.

Preuve.

$$\lambda(t) = \beta \frac{(t - \gamma)^{(\beta-1)}}{\sigma^\beta} \text{ pour tout réel } t \text{ de son domaine de définition}$$

Sens de variation

Pour $\beta = 1$

Pour tout réel $t > \gamma$, $(t - \gamma)^{(\beta-1)} = 1$ et donc $\lambda = \frac{1}{\sigma}$. Par continuité, $\lambda = \frac{1}{\sigma}$ pour $t \geq \gamma$.

Pour $\beta \neq 1$

La fonction $\lambda : t \mapsto \beta \frac{(t - \gamma)^{(\beta-1)}}{\sigma^\beta}$ est dérivable en tout réel $t > \gamma$ et on a

$$\frac{d}{dt} \lambda(t) = \beta(\beta - 1) \frac{(t - \gamma)^{(\beta-2)}}{\sigma^\beta} \quad (\text{B.4.5})$$

Ainsi :

Si $\beta \in]0, 1[$, alors $\frac{d}{dt} \lambda(t) < 0$ pour tout réel $t > \gamma$. La fonction λ est donc strictement décroissante.

Si $\beta \in]1, +\infty[$, alors $\frac{d}{dt}\lambda(t) > 0$ pour tout réel $t > \gamma$. La fonction λ est donc strictement croissante. Cela se prolonge par continuité en $t = \gamma$.

Convexité

Par ailleurs :

Si $\beta = 2$, alors pour tout réel $t > \gamma$, $\frac{d}{dt}\lambda(t) = \frac{\beta(\beta-1)}{\sigma^\beta} = \frac{2}{\sigma^2}$. λ est donc une fonction linéaire dont la pente est $\frac{2}{\sigma^2}$.

Si $\beta \neq 2$, alors la dérivée seconde de λ , qui est définie en tout réel $t > \gamma$, est donnée par

$$\frac{d^2}{dt^2}\lambda(t) = \beta(\beta-1)(\beta-2)\frac{(t-\gamma)^{(\beta-3)}}{\sigma^\beta} \quad (\text{B.4.6})$$

Et comme pour tout réel $t > \gamma$, $\beta\frac{(t-\gamma)^{(\beta-3)}}{\sigma^\beta} > 0$, alors :

Si $\beta \in]1, 2[$, alors $\frac{d^2}{dt^2}\lambda(t) < 0$ pour tout réel $t > \gamma$. λ est donc une fonction concave. Cela se prolonge par continuité en $t = \gamma$.

Si $\beta \in]0, 1[$ ou $\beta \in]2, +\infty[$, alors $\frac{d^2}{dt^2}\lambda(t) > 0$ pour tout réel $t > \gamma$. λ est donc une fonction convexe. Pour $\beta \in]2, +\infty[$, cela se prolonge par continuité en $t = \gamma$.

Asymptotes et axes paraboliques

Enfin :

Si $\beta \in]0, 1[$, alors

$$\lim_{t \rightarrow +\infty} \lambda(t) = 0 \quad (\text{B.4.7})$$

Si $\beta > 1$, alors

$$\lim_{t \rightarrow +\infty} \lambda(t) = +\infty \quad (\text{B.4.8})$$

Et l'on a

$$\begin{aligned} \lim_{t \rightarrow +\infty} \frac{\lambda(t)}{t} &= \lim_{t \rightarrow +\infty} \frac{\beta}{\sigma^\beta} \frac{t^{(\beta-1)}}{t} \\ &= \lim_{t \rightarrow +\infty} \frac{\beta}{\sigma^\beta} t^{(\beta-2)} \end{aligned} \quad (\text{B.4.9})$$

Donc :

Si $\beta \in]1, 2[$, alors $\lim_{t \rightarrow +\infty} \frac{\lambda(t)}{t} = 0$, d'où l'on a la branche parabolique de direction l'axe des temps t .

Si $\beta > 2$, alors $\lim_{t \rightarrow +\infty} \frac{\lambda(t)}{t} = +\infty$, d'où l'on a la branche parabolique de direction l'axes de λ .

Le cas où $\beta = 2$, est déjà réglé, vu que dans ce cas, λ est linéaire. \square

Connaissant le comportement du taux de défaillance λ , revenons à la fonction disponibilité qui est l'objet de ce chapitre.

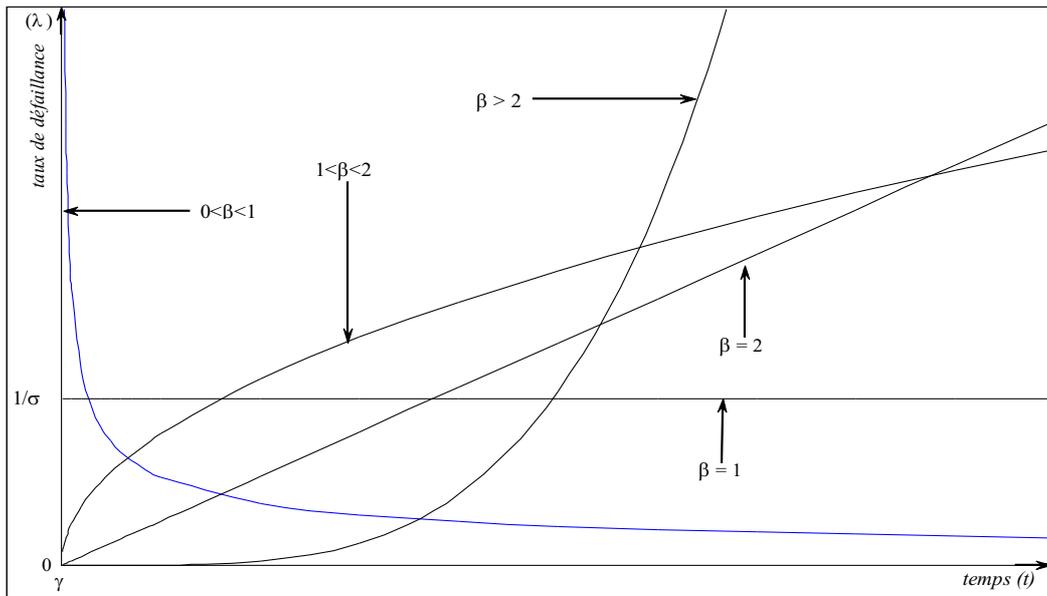


Figure B.1 – Évolution du taux de défaillance suivant les valeurs du paramètre de forme

B.5 La fonction Disponibilité dans le cas d’une loi de Weibull

Dans cette section, nous allons dans un premier temps voir la fonction disponibilité d’une entité lorsque son taux de réparation est défini de façon général en fonction du temps, et dans un second temps et de façon très détaillé, cette fonction disponibilité lorsque le taux de réparation est constante.

B.5.1 Loi de Weibull et taux de réparation général

D’après la section précédente, le taux de défaillance λ d’une entité dont la fiabilité est soumise à une loi de Weibull est définie par

$$\begin{aligned} \lambda : [\max(0, \gamma), +\infty[&\longrightarrow \mathbb{R}_+ \\ t &\longmapsto \lambda(t) = \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \end{aligned}$$

où $\gamma \in \mathbb{R}$ est le paramètre de décalage, $\sigma \in \mathbb{R}_+^*$ le paramètre d’échelle, et $\beta \in \mathbb{R}_+^*$ le paramètre de forme.

Nous supposons que son taux de réparation peut être défini en fonction du temps par

$$\begin{aligned} \mu : [\max(0, \gamma), +\infty[&\longrightarrow \mathbb{R}_+^* \\ t &\longmapsto \mu(t) \end{aligned} \tag{B.5.1}$$

Théorème B.5.1.1. Équation différentielle de la disponibilité dans le cas de la loi de Weibull

Soit une entité E dont la fiabilité suit une loi de Weibull et dont le taux de défaillance $\lambda : t \longmapsto \lambda(t)$ et le taux de réparation $\mu : t \longmapsto \mu(t)$ sont respectivement définis dans les expressions B.5.1 et B.5.1.

Alors sa fonction disponibilité $A \mapsto A(t)$ vérifie l'équation différentielle

$$\frac{d}{dt}A(t) + \left(\mu(t) + \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \right) A(t) = \mu(t) \quad (\text{B.5.2})$$

Cette équation peut être établie directement à partir de l'expression du taux de défaillance dans le cas de la loi de Weibull et du taux de réparation, ou bien être déduite de l'équation dans le cas général B.2.1.

Preuve. Les taux de défaillance et de réparation dans le cas du théorème B.5.1.1 vérifient bien les hypothèses du théorème B.2.1.1 établissant l'équation différentielle de la fonction disponibilité dans le cas général. Ainsi, le théorème précédent n'est autre chose que le théorème B.2.1.1 dans lequel le taux de défaillance est remplacé par son expression qui est connue. \square

L'expression de la fonction disponibilité dans ce cas peut ainsi être retrouvée en résolvant l'équation B.5.2 ou bien déduite de l'expression B.2.6 de la disponibilité dans le cas général dans le cas général.

Théorème B.5.1.2. Expression de la disponibilité dans le cas de la loi de Weibull

Soit une entité E dont la fiabilité suit une loi de Weibull et dont le taux de défaillance $\lambda : t \mapsto \lambda(t)$ et le taux de réparation $\mu : t \mapsto \mu(t)$ sont respectivement définis dans les expressions B.5.1 et B.5.1 avec un paramètre de décalage $\gamma \in \mathbb{R}$, un paramètre d'échelle $\sigma \in \mathbb{R}_+^*$, et un paramètre de forme $\beta \in \mathbb{R}_+^*$. Supposons que le taux de réparation soit intégrable sur toute partie bornée de \mathbb{R} .

Alors sa fonction disponibilité $A \mapsto A(t)$ est définie par :

$$A : [\max(0, \gamma), +\infty[\longrightarrow [0, 1] \\ t \longmapsto A(t)$$

avec

$$A(t) = \exp \left(- \left[\int_\gamma^t \mu(x) dx + \left(\frac{t-\gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \int_\gamma^t \mu(x) \cdot \exp \left(\int_\gamma^x \mu(u) du + \left(\frac{x-\gamma}{\sigma} \right)^\beta \right) dx \right\} \quad (\text{B.5.3})$$

Preuve. Pour démontrer ce théorème, nous allons nous baser sur l'expression de la fonction disponibilité donnée dans le cas général dans le théorème B.2.2.1.

Rappelons l'expression du taux de défaillance défini par

$$\lambda(t) = \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta} \quad (\text{B.5.4})$$

Lorsque le paramètre de forme $\beta \geq 1$, le taux de défaillance λ tel que rappelé dans l'expression B.5.4 est une fonction bornée sur toute partie bornée de \mathbb{R} (supposant qu'elle vaut 0 lorsque $t < \gamma$). C'est donc une fonction intégrable sur toute partie bornée de \mathbb{R} .

Dans le cas où $0 < \beta < 1$, le taux de défaillance est aussi intégrable sur toute partie bornée de \mathbb{R} . En effet, pour $\varepsilon \in \mathbb{R}_+^*$, la fonction λ est bornée sur toute partie bornée de l'ensemble $] -\infty, \gamma[\cup] \gamma + \varepsilon, +\infty[$. Elle a une limite infinie lorsque t tend vers γ^+ . Mais une de ses primitives est

la fonction $t \mapsto \left(\frac{t-\gamma}{\sigma}\right)^\beta$. Cette primitive est une fonction bornée sur toute partie bornée de \mathbb{R} (étant définie comme 0 lorsque $t < \gamma$). Le taux de défaillance est donc intégrable sur toute partie bornée de \mathbb{R} .

Par ailleurs, le taux de réparation est intégrable sur toute partie bornée de \mathbb{R} . Les hypothèses du théorème B.2.2.1 sont donc vérifiées et le réel a tel que $A(a) = 1$ est le paramètre de décalage γ .

Ainsi, d'après le théorème B.2.2.1, la fonction disponibilité est définie pour $t \geq \gamma$ par

$$\begin{aligned}
A(t) &= \exp\left(-\int_\gamma^t \mu(x)dx - \int_\gamma^t \lambda(x)dx\right) \cdot \left\{1 + \int_\gamma^t \mu(x) \cdot \exp\left(\int_\gamma^x \mu(u)du + \int_\gamma^x \lambda(u)du\right) dx\right\} \\
&= \exp\left(-\int_\gamma^t \mu(x)dx - \int_\gamma^t \beta \frac{(x-\gamma)^{(\beta-1)}}{\sigma^\beta} dx\right) \cdot \\
&\quad \left\{1 + \int_\gamma^t \mu(x) \cdot \exp\left(\int_\gamma^x \mu(u)du + \int_\gamma^x \beta \frac{(u-\gamma)^{(\beta-1)}}{\sigma^\beta} du\right) dx\right\} \\
&= \exp\left(-\int_\gamma^t \mu(x)dx - \left[\left(\frac{x-\gamma}{\sigma}\right)^\beta\right]_\gamma^t\right) \cdot \\
&\quad \left\{1 + \int_\gamma^t \mu(x) \cdot \exp\left(\int_\gamma^x \mu(u)du + \left[\left(\frac{u-\gamma}{\sigma}\right)^\beta\right]_\gamma^x\right) dx\right\} \\
&= \exp\left(-\int_\gamma^t \mu(x)dx - \left(\frac{t-\gamma}{\sigma}\right)^\beta\right) \cdot \left\{1 + \int_\gamma^t \mu(x) \cdot \exp\left(\int_\gamma^x \mu(u)du + \left(\frac{x-\gamma}{\sigma}\right)^\beta\right) dx\right\}
\end{aligned} \tag{B.5.5}$$

□

La sous-section suivante est consacrée à l'étude en détail de cette fonction disponibilité dans le cas de la loi de Weibull lorsque le taux de réparation μ est constant.

B.5.2 Loi de Weibull et taux de réparation constant

Le corollaire suivant donne l'équation différentielle de la fonction disponibilité d'une entité dans le cas où la fiabilité suit une loi de Weibull et lorsque le taux de réparation est constant. Comme dans les autres cas, cette équation différentielle peut-être établie indépendamment du cas général.

Corollaire B.5.2.1. Équation différentielle de la disponibilité dans le cas de la loi de Weibull lorsque le taux de défaillance est constant

Soit une entité E dont la fiabilité suit une loi de Weibull et dont le taux de défaillance $\lambda : t \mapsto \lambda(t)$ est défini dans l'expression B.5.1. Supposons que son taux de réparation $\mu > 0$ soit constant.

Alors sa fonction disponibilité $A \mapsto A(t)$ vérifie l'équation différentielle

$$\frac{d}{dt}A(t) + \left(\mu + \beta \frac{(t-\gamma)^{(\beta-1)}}{\sigma^\beta}\right) A(t) = \mu \tag{B.5.6}$$

La preuve de ce corollaire découle directement du théorème B.5.1.1.

Dans ce cas où le taux de réparation est constant, la fonction disponibilité de l'entité est alors donnée dans le corollaire suivant.

Corollaire B.5.2.2. Expression de la disponibilité dans le cas de la loi de Weibull lorsque le taux de réparation est constant

Soit une entité E dont la fiabilité suit une loi de Weibull et dont le taux de défaillance $\lambda : t \mapsto \lambda(t)$ est défini dans l'expression B.5.1 avec un paramètre de décalage $\gamma \in \mathbb{R}$, un paramètre d'échelle $\sigma \in \mathbb{R}_+^*$, et un paramètre de forme $\beta \in \mathbb{R}_+^*$. Supposons que le taux de réparation $\mu > 0$ de cette entité soit constant.

Alors sa fonction disponibilité $A : t \mapsto A(t)$ est définie par :

$$A : [\max(0, \gamma), +\infty[\longrightarrow [0, 1] \\ t \longmapsto A(t)$$

avec

$$A(t) = \exp \left(- \left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \mu \sigma \cdot \int_0^{\frac{t - \gamma}{\sigma}} \exp(\mu \sigma x + x^\beta) dx \right\} \quad (\text{B.5.7})$$

Preuve. Reprenons l'expression de la fonction disponibilité dans le cas de la loi de Weibull dans le cas général du taux de réparation telle qu'elle est dans le théorème B.5.1.2 pour le cas où le taux de réparation est constant.

$$A(t) = \exp \left(- \left[\int_\gamma^t \mu dx + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \int_\gamma^t \mu \cdot \exp \left(\int_\gamma^x \mu du + \left(\frac{x - \gamma}{\sigma} \right)^\beta \right) dx \right\} \\ = \exp \left(- \left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \mu \cdot \int_\gamma^t \exp \left(\mu \cdot (x - \gamma) + \left(\frac{x - \gamma}{\sigma} \right)^\beta \right) dx \right\} \quad (\text{B.5.8})$$

Procédons alors au changement de variable suivant dans la deuxième ligne de l'équation B.5.8.

$$u = \frac{x - \gamma}{\sigma} \text{ ou encore } x = \sigma \cdot u + \gamma \text{ qui donne } dx = \sigma \cdot du$$

Alors

$$A(t) = \exp \left(- \left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \cdot \left\{ 1 + \mu \sigma \cdot \int_0^{\frac{t - \gamma}{\sigma}} \exp(\mu \cdot u + u^\beta) du \right\} \quad (\text{B.5.9})$$

□

Dans la suite de ce chapitre, nous allons étudier les différentes caractéristiques de la fonction disponibilité dans le cas du théorème précédent.

Lemme B.5.2.3. Fonction simplifiée pour l'étude de la disponibilité

Soit σ , β , γ et μ des réels tels que $\sigma > 0$, $\beta > 0$, $\gamma \in \mathbb{R}$ et $\mu > 0$.

Soit les fonctions A et A^1 définies par:

$$A : [\gamma, +\infty[\longrightarrow [0, 1] \quad A^1 : \mathbb{R}_+ \longrightarrow [0, 1] \\ t \longmapsto A(t) \quad t \longmapsto A^1(t)$$

avec

$$A(t) = \exp\left(-\left[\mu \cdot (t - \gamma) + \left(\frac{t - \gamma}{\sigma}\right)^\beta\right]\right) \cdot \left\{1 + \mu \cdot \sigma \cdot \int_0^{\frac{t - \gamma}{\sigma}} \exp(\mu \sigma x + x^\beta) dx\right\} \quad (\text{B.5.10})$$

$$A^1(t) = \exp\left(-\left[\mu \cdot \sigma \cdot t + t^\beta\right]\right) \cdot \left\{1 + \mu \sigma \int_0^t \exp(\mu \sigma x + x^\beta) dx\right\} \quad (\text{B.5.11})$$

Alors :

a) Si A^1 est non croissante (resp. non décroissante) sur une partie U de \mathbb{R} , alors A est non croissante (resp. non décroissante) sur l'image de U par la fonction $t \mapsto (\sigma t + \gamma)$.

b) En plus la limite lorsque t tend vers plus l'infinie de $A(t)$ est égale à la limite lorsque t tend vers plus l'infinie de $A^1(t)$.

Preuve. Par changement de variable, on obtient

$$A(t) = A^1\left(\frac{t - \gamma}{\sigma}\right) \text{ ou encore } A^1(t) = A(\sigma t + \gamma) \quad (\text{B.5.12})$$

Or la fonction $t \mapsto \frac{t - \gamma}{\sigma}$ est une fonction non décroissante. Cela permet l'assertion a). De plus cette fonction tend vers plus l'infinie lorsque t tend vers plus l'infinie justifiant ainsi l'assertion b). \square

Ce lemme permet d'étudier le comportement de la fonction A par l'intermédiaire de la fonction A^1 . Ainsi dans toutes les études pour analyser le comportement de la fonction de disponibilité $t \mapsto A(t)$, nous considérerons dans la plupart des cas plutôt la fonction $t \mapsto A^1(t)$.

Le paramètre de forme β est en réalité le seul paramètre influant sur le comportement du taux de défaillance, de la fiabilité, et de la disponibilité de l'entité dans le cas de la loi de Weibull. Le paramètre de décalage γ est juste une origine temporelle, et le paramètre d'échelle σ constitue juste un facteur d'échelle comme son nom l'indique. Nous allons donc étudier les différentes caractéristiques de la fonction disponibilité $t \mapsto A(t)$ suivant les variations du paramètre de forme β . Nous noterons A_β la fonction disponibilité pour une valeur donnée du paramètre de forme β .

Remarque B.5.2.4. Dans ce qui suit et en particulier dans les théorèmes B.5.2.5 et B.5.2.6, nous supposons que la fonction disponibilité dans le cas considéré ici admet une limite lorsque le temps t tend vers plus l'infinie. Bien sûr intuitivement, le contraire serait plutôt surprenant. Nous apporterons la preuve de cette affirmation un peu plus loin.

Le théorème suivant donne les relations entre les disponibilités asymptotiques dans le cas où la fiabilité suit une loi de Weibull, suivant les valeurs du paramètre de forme β .

Théorème B.5.2.5. La disponibilité asymptotique suivant les valeurs du paramètre de forme
Soit σ , β et γ ($\sigma > 0$, $\beta > 0$ et $\gamma \in \mathbb{R}$) respectivement les paramètres d'échelle, de forme et de position de la loi de Weibull. Soit μ ($\mu > 0$) le taux de réparation de l'entité. Supposons que ce taux de réparation μ soit constant. Posons $A_\beta(\infty)$ la limite lorsque t tend vers plus l'infinie de la fonction disponibilité A_β qui à t associe $A_\beta(t)$ pour des valeurs de σ , γ , β et μ fixées.

Alors la fonction qui à β associe $A_\beta(\infty)$ est une fonction strictement décroissante.

Cette observation justifie ce que l'on aurait pu penser intuitivement. En effet, puisque le taux de défaillance se dégrade de plus en plus vite lorsque le paramètre de forme β croît, cela amène à imaginer que la disponibilité de l'entité se dégrade aussi de plus en plus vite et de plus en plus profondément.

Preuve. La fonction disponibilité, dans les conditions du théorème, est donnée par la restriction à \mathbb{R}_+ de

$$\begin{aligned} A_\beta : [\gamma, +\infty[&\longrightarrow [0,1] \\ t &\longmapsto A_\beta(t) = \exp\left(-\left[\mu(t-\gamma) + \left(\frac{t-\gamma}{\sigma}\right)^\beta\right]\right) \cdot \left\{1 + \mu\sigma \int_0^{\frac{t-\gamma}{\sigma}} \exp(\mu\sigma x + x^\beta) dx\right\} \end{aligned} \quad (\text{B.5.13})$$

En utilisant le lemme B.5.2.3, nous allons, à la place de cette fonction, plutôt étudier le comportement de la limite lorsque t tend vers plus l'infinie, notée $A_\beta^1(\infty)$ de la fonction

$$\begin{aligned} A_\beta^1 : [0, +\infty[&\longrightarrow [0,1] \\ t &\longmapsto A_\beta^1(t) = \left\{\exp\left(-\left[\mu\sigma t + t^\beta\right]\right)\right\} \cdot \left\{1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx\right\} \end{aligned} \quad (\text{B.5.14})$$

Elle peut se réécrire

$$A_\beta^1(t) = \exp\left(-\left[\mu\sigma t + t^\beta\right]\right) + \mu\sigma \int_0^t \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right) dx \quad (\text{B.5.15})$$

Comme $\lim_{t \rightarrow +\infty} \exp\left(-\left[\mu\sigma t + t^\beta\right]\right) = 0$, alors

$$\begin{aligned} \lim_{t \rightarrow +\infty} A_\beta^1(t) &= \mu\sigma \cdot \lim_{t \rightarrow +\infty} \int_0^t \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right) dx \\ &= \mu\sigma \cdot \lim_{t \rightarrow +\infty} \left\{ \int_0^1 \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right) dx + \int_1^t \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right) dx \right\} \\ &= \mu\sigma \cdot \lim_{t \rightarrow +\infty} \int_1^t \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right) dx \end{aligned} \quad (\text{B.5.16})$$

La dernière égalité provient du fait que la fonction $x \longmapsto \exp\left(-\mu\sigma(t-x) - t^\beta + x^\beta\right)$ converge simplement vers 0 sur l'intervalle $[0,1]$, lorsque t tend vers plus l'infinie.

Par ailleurs, soit, pour x et t fixés tels que $1 \leq x \leq t$, la fonction $f_{x,t}$ définie par

$$\begin{aligned} f_{x,t} : \mathbb{R}_+^* &\longrightarrow \mathbb{R}_+^* \\ \beta &\longmapsto f_{x,t}(\beta) = \exp(-\mu\sigma(t-x)) \cdot \exp\left(-t^\beta + x^\beta\right) \end{aligned} \quad (\text{B.5.17})$$

ou encore

$$f_{x,t}(\beta) = \exp(-\mu\sigma(t-x)) \cdot \exp\left(-t^\beta\right) \cdot \exp\left(x^\beta\right) \quad (\text{B.5.18})$$

La fonction $f_{x,t}$ est dérivable sur \mathbb{R}_+^* et sa dérivée est donnée par

$$\frac{d}{d\beta} f_{x,t}(\beta) = \exp(-\mu\sigma(t-x)) \cdot \left\{ x^\beta \ln(x) \exp\left(-t^\beta\right) \cdot \exp\left(x^\beta\right) - t^\beta \ln(t) \exp\left(-t^\beta\right) \cdot \exp\left(x^\beta\right) \right\}$$

soit

$$\frac{d}{d\beta} f_{x,t}(\beta) = \exp(-\mu\sigma(t-x)) \cdot \exp(-t^\beta) \cdot \exp(x^\beta) \cdot (x^\beta \ln(x) - t^\beta \ln(t)) \quad (\text{B.5.19})$$

Le signe de cette dérivée est celui de la fonction

$$\begin{aligned} g : \mathbb{R}_+^* &\longrightarrow \mathbb{R} \\ \beta &\longmapsto g(\beta) = (x^\beta \ln(x) - t^\beta \ln(t)) \end{aligned} \quad (\text{B.5.20})$$

Or la fonction $x \longmapsto x^\beta \ln(x)$ (pour $\beta > 0$ fixé) est dérivable sur \mathbb{R}_+^* et sa fonction dérivée est $x \longmapsto x^{\beta-1}(1 + \beta \ln(x))$. Cette dernière fonction est positive en tout $x \geq 1$ - cela est vrai en fait dès que l'on a $x \geq \exp\left(\frac{-1}{\beta}\right)$. Donc la fonction $x \longmapsto x^\beta \ln(x)$ est une fonction non décroissante sur $[1, +\infty[$. Ainsi, pour $t \geq x \geq 1$, nous avons $x^\beta \ln(x) \leq t^\beta \ln(t)$. La fonction g est donc négative, et par suite, $\frac{d}{d\beta} f_{x,t}(\beta) \leq 0$ pour tout $\beta > 0$. La fonction $f_{x,t}$ est donc décroissante.

Soit alors deux réels β_1 et β_2 tels que $0 < \beta_1 \leq \beta_2$. Pour tout réel $t \geq 1$ et pour tout réel x tel que $1 \leq x \leq t$, nous avons

$$f_{x,t}(\beta_1) \geq f_{x,t}(\beta_2) \quad (\text{B.5.21})$$

ou encore

$$\exp\left(-\mu\sigma(t-x) - t^{\beta_1} + x^{\beta_1}\right) \geq \exp\left(-\mu\sigma(t-x) - t^{\beta_2} + x^{\beta_2}\right) \quad (\text{B.5.22})$$

et donc

$$\int_1^t \exp\left(-\mu\sigma(t-x) - t^{\beta_1} + x^{\beta_1}\right) dx \geq \int_1^t \exp\left(-\mu\sigma(t-x) - t^{\beta_2} + x^{\beta_2}\right) dx \quad (\text{B.5.23})$$

Par conséquent, à la limite, nous avons

$$\lim_{t \rightarrow +\infty} \int_1^t \exp\left(-\mu\sigma(t-x) - t^{\beta_1} + x^{\beta_1}\right) dx \geq \lim_{t \rightarrow +\infty} \int_1^t \exp\left(-\mu\sigma(t-x) - t^{\beta_2} + x^{\beta_2}\right) dx \quad (\text{B.5.24})$$

Cela signifie, compte tenue des égalités B.5.16, que

$$A_{\beta_1}^1(\infty) \geq A_{\beta_2}^1(\infty) \quad (\text{B.5.25})$$

Cette dernière égalité revient, compte tenu du lemme B.5.2.3, à

$$A_{\beta_1}(\infty) \geq A_{\beta_2}(\infty) \quad (\text{B.5.26})$$

□

Par ailleurs, pour des valeurs données du taux de réparation (μ), du paramètre d'échelle (σ), du paramètre de forme (β), un bon encadrement de la limite lorsque le temps t tend vers plus l'infinie de la fonction disponibilité dans le cas où la fiabilité suit une loi de Weibull, peut être donné par le théorème suivant. Comme cet encadrement dépend du paramètre de forme β , nous allons toujours garder la notation du théorème B.5.2.5.

Théorème B.5.2.6. Encadrement et estimation de la disponibilité asymptotique

Soit σ , β et γ ($\sigma > 0$, $\beta > 0$ et $\gamma \in \mathbb{R}$) respectivement les paramètres d'échelle, de forme et de position de la loi de Weibull. Soit μ ($\mu > 0$) le taux de réparation de l'entité. Supposons que ce taux de réparation μ soit constant. Posons $A_\beta(\infty)$ la limite lorsque t tend vers plus l'infinie de la fonction disponibilité A_β qui à t associe $A_\beta(t)$ pour des valeurs de σ , γ , β et μ fixées. Alors :

- a) Si $\beta = 1$, alors $A_\beta(\infty) = \frac{\mu}{\mu + \frac{1}{\sigma}}$.
- b) Si $0 < \beta < 1$ alors $\frac{\mu}{\mu + \frac{\beta}{\sigma}} < A_\beta(\infty) \leq 1$.
- c) Si $\beta > 1$ alors $0 \leq A_\beta(\infty) < \frac{\mu}{\mu + \frac{\beta}{\sigma}}$.

Le premier cas ($\beta = 1$) correspond au cas où la loi de Weibull se ramène à la loi exponentielle avec un taux de défaillance qui vaut $\frac{1}{\sigma}$. Ce cas peut être étudié directement en considérant la disponibilité avec une loi exponentielle, ou bien être traité comme un cas particulier du cas général étudié ici.

Remarque B.5.2.7. *Compte tenu du théorème B.5.2.5, et du fait que $0 \leq A_\beta(\infty) \leq 1$, on peut déjà remarquer que*

- a) Lorsque $0 < \beta < 1$, on a $A_1(\infty) \leq A_\beta(\infty) \leq 1$ (avec $A_1(\infty) = \frac{\mu}{\mu + \frac{1}{\sigma}}$).
- b) Lorsque $\beta > 1$, on a $0 \leq A_\beta(\infty) \leq A_1(\infty)$.

Mais le théorème B.5.2.6 apporte largement plus d'information que cette remarque.

Preuve. Nous rappelons que la fonction disponibilité est définie par

$$A_\beta : \mathbb{R} \longrightarrow [0,1]$$

$$t \longmapsto \begin{cases} A_\beta(t) = \exp \left(- \left[\mu(t - \gamma) + \left(\frac{t - \gamma}{\sigma} \right)^\beta \right] \right) \\ \left\{ 1 + \mu\sigma \int_0^{\frac{t-\gamma}{\sigma}} \exp(\mu\sigma x + x^\beta) dx \right\} \end{cases} \quad (\text{B.5.27})$$

Mais tout comme pour la preuve du théorème B.5.2.5, nous allons, compte tenu du lemme B.5.2.3, utiliser la fonction A_β^1 dont nous rappelons la définition ci-dessous.

$$A_\beta^1 : \mathbb{R}_+ \longrightarrow [0,1]$$

$$t \longmapsto A_\beta^1(t) = \exp \left(- \left[\mu\sigma t + t^\beta \right] \right) + \mu\sigma \int_0^t \exp \left(-\mu\sigma(t-x) - t^\beta + x^\beta \right) dx \quad (\text{B.5.28})$$

Tout comme dans l'équation B.5.16, la limite quand t tend vers plus l'infinie de $A_\beta^1(t)$, est donnée par

$$\lim_{t \rightarrow +\infty} A_\beta^1(t) = \mu\sigma \cdot \lim_{t \rightarrow +\infty} \int_1^t \exp \left(-\mu\sigma(t-x) - t^\beta + x^\beta \right) dx \quad (\text{B.5.29})$$

Posons donc, pour $t \geq 1$,

$$I_\beta(t) = \int_1^t \exp \left(-\mu\sigma(t-x) - t^\beta + x^\beta \right) dx$$

$$= \exp \left(-\mu\sigma t - t^\beta \right) \int_1^t \exp \left(\mu\sigma x + x^\beta \right) dx \quad (\text{B.5.30})$$

Alors

$$\lim_{t \rightarrow +\infty} A_\beta^1(t) = \mu\sigma \lim_{t \rightarrow +\infty} I_\beta(t) \quad (\text{B.5.31})$$

Nous allons donc étudier la limite lorsque t tend vers plus l'infinie de $I_\beta(t)$ suivant les trois cas du théorème.

Premier cas : $\beta = 1$

Dans ce cas, on peut calculer aisément $I_\beta(t)$ qui donne

$$\begin{aligned} I_\beta(t) &= \exp(-(\mu\sigma + 1)t) \int_1^t \exp((\mu\sigma + 1)x) dx \\ &= \frac{1}{\mu\sigma + 1} \exp(-(\mu\sigma + 1)t) [\exp((\mu\sigma + 1)x)]_1^t \\ &= \frac{1}{\mu\sigma + 1} \exp(-(\mu\sigma + 1)t) \{ \exp((\mu\sigma + 1)t) - \exp(\mu\sigma + 1) \} \\ &= \frac{1}{\mu\sigma + 1} \{ 1 - \exp((\mu\sigma + 1)(1 - t)) \} \end{aligned} \quad (\text{B.5.32})$$

Et comme

$$\lim_{t \rightarrow +\infty} \{ \exp((\mu\sigma + 1)(1 - t)) \} = 0 \quad (\text{B.5.33})$$

alors

$$\lim_{t \rightarrow +\infty} I_\beta(t) = \frac{1}{\mu\sigma + 1} \quad (\text{B.5.34})$$

Ainsi, en utilisant l'égalité B.5.31, nous retrouvons

$$\lim_{t \rightarrow +\infty} A_\beta^1(t) = \frac{\mu}{\mu + \frac{1}{\sigma}} \quad (\text{B.5.35})$$

Deuxième cas : $0 < \beta < 1$

Dans ce deuxième cas, l'on a $-1 < \beta - 1 < 0$. Ainsi, pour tout réel x tel que $x \geq 1$, nous avons

$$0 < x^{\beta-1} \leq 1 \quad (\text{B.5.36})$$

de sorte que

$$\mu\sigma \leq \mu\sigma + \beta x^{\beta-1} \leq \mu\sigma + \beta \quad (\text{B.5.37})$$

Cela implique, en utilisant la dernière de cette chaîne d'inégalités que

$$\begin{aligned} &\exp(-\mu\sigma t - t^\beta) \cdot \left\{ \int_1^t (\mu\sigma + \beta x^{\beta-1}) \exp(\mu\sigma x + x^\beta) dx \right\} \\ &\leq (\mu\sigma + \beta) \cdot \exp(-\mu\sigma t - t^\beta) \cdot \left\{ \int_1^t \exp(\mu\sigma x + x^\beta) dx \right\} \end{aligned} \quad (\text{B.5.38})$$

Ce qui veut dire que

$$\exp(-\mu\sigma t - t^\beta) \int_1^t (\mu\sigma + \beta x^{\beta-1}) \exp(\mu\sigma x + x^\beta) dx \leq (\mu\sigma + \beta) I_\beta(t) \quad (\text{B.5.39})$$

Cela revient, en intégrant l'équation du premier membre de cette inégalité, à

$$\exp(-\mu\sigma t - t^\beta) \left[\exp(\mu\sigma x + x^\beta) \right]_1^t \leq (\mu\sigma + \beta) I_\beta(t) \quad (\text{B.5.40})$$

c'est à dire

$$\left\{ \exp(-\mu\sigma t - t^\beta) \right\} \left\{ \exp(\mu\sigma t + t^\beta) - \exp(\mu\sigma + 1) \right\} \leq (\mu\sigma + \beta) I_\beta(t) \quad (\text{B.5.41})$$

ou encore

$$\frac{1}{\mu\sigma + \beta} \left\{ 1 - \exp\left((\mu\sigma + 1) - (\mu\sigma t + t^\beta)\right) \right\} \leq I_\beta(t) \quad (\text{B.5.42})$$

Et comme

$$\lim_{t \rightarrow +\infty} \exp\left((\mu\sigma + 1) - (\mu\sigma t + t^\beta)\right) = 0 \quad (\text{B.5.43})$$

alors à la limite, nous avons

$$\frac{1}{\mu\sigma + \beta} \leq \lim_{t \rightarrow +\infty} I_\beta(t) \quad (\text{B.5.44})$$

D'autre part, pour tout réel β tel que $\beta > 0$, la fonction définie par $x \mapsto x^\beta$ est une fonction croissante sur \mathbb{R}_+ . Ainsi, nous avons

$$\int_1^t \exp(\mu\sigma x + x^\beta) dx \leq \int_1^t \exp(\mu\sigma x + t^\beta) dx \quad (\text{B.5.45})$$

ce qui signifie que

$$I_\beta(t) \leq \exp(-\mu\sigma t - t^\beta) \int_1^t \exp(\mu\sigma x + t^\beta) dx \quad (\text{B.5.46})$$

Or

$$\begin{aligned} \exp(-\mu\sigma t - t^\beta) \int_1^t \exp(\mu\sigma x + t^\beta) dx &= \exp(-\mu\sigma t) \int_1^t \exp(\mu\sigma x) dx \\ &= \frac{1}{\mu\sigma} \exp(-\mu\sigma t) [\exp(\mu\sigma t) - \exp(\mu\sigma)] \\ &= \frac{1}{\mu\sigma} [1 - \exp(\mu\sigma(1 - t))] \end{aligned} \quad (\text{B.5.47})$$

Donc

$$I_\beta(t) \leq \frac{1}{\mu\sigma} [1 - \exp(\mu\sigma(1 - t))] \quad (\text{B.5.48})$$

Et puisque

$$\lim_{t \rightarrow +\infty} \exp(\mu\sigma(1 - t)) = 0 \quad (\text{B.5.49})$$

alors on a à la limite

$$\lim_{t \rightarrow +\infty} I_\beta(t) \leq \frac{1}{\mu\sigma} \quad (\text{B.5.50})$$

Ainsi, avec les inégalités B.5.44 et B.5.50, nous avons

$$\frac{1}{\mu\sigma + \beta} \leq \lim_{t \rightarrow +\infty} I_\beta(t) \leq \frac{1}{\mu\sigma} \quad (\text{B.5.51})$$

Finalement, en utilisant cette chaîne d'inégalités B.5.51 et l'égalité B.5.31, nous retrouvons

$$\frac{\mu}{\mu + \frac{\beta}{\sigma}} < \lim_{t \rightarrow +\infty} A_\beta^1(t) \leq 1 \quad (\text{B.5.52})$$

Troisième cas : $\beta > 1$

Pour ce troisième cas, nous avons $\frac{1}{\beta} < 1$ et donc $\frac{1}{\beta} - 1 < 0$, si bien que pour tout réel u tel que $u \geq 1$, nous avons

$$u^{\frac{1}{\beta} - 1} \leq 1 \quad (\text{B.5.53})$$

de sorte que

$$1 + \frac{\mu\sigma}{\beta} u^{\left(\frac{1}{\beta}-1\right)} \leq 1 + \frac{\mu\sigma}{\beta} \quad (\text{B.5.54})$$

Donc si nous posons pour $t \geq 1$

$$J(t) = \exp(-\mu\sigma t - t^\beta) \int_1^{t^\beta} \exp\left(u + \mu\sigma u^{\frac{1}{\beta}}\right) du \quad (\text{B.5.55})$$

nous avons

$$\exp(-\mu\sigma t - t^\beta) \int_1^{t^\beta} \left(1 + \frac{\mu\sigma}{\beta} u^{\left(\frac{1}{\beta}-1\right)}\right) \exp\left(u + \mu\sigma u^{\frac{1}{\beta}}\right) du \leq \left(1 + \frac{\mu\sigma}{\beta}\right) J(t) \quad (\text{B.5.56})$$

En intégrant l'équation du premier membre, cela revient à

$$\exp(-\mu\sigma t - t^\beta) \left[\exp\left(u + \mu\sigma u^{\frac{1}{\beta}}\right)\right]_1^{t^\beta} \leq \left(1 + \frac{\mu\sigma}{\beta}\right) J(t) \quad (\text{B.5.57})$$

ce qui donne

$$\exp(-\mu\sigma t - t^\beta) \left\{ \exp\left(t^\beta + \mu\sigma t\right) - \exp(1 + \mu\sigma) \right\} \leq \left(\frac{\mu\sigma}{\beta} + 1\right) J(t) \quad (\text{B.5.58})$$

soit

$$\frac{\beta}{\mu\sigma + \beta} \left\{ 1 - \exp\left((1 + \mu\sigma) - (\mu\sigma t + t^\beta)\right) \right\} \leq J(t) \quad (\text{B.5.59})$$

À la limite, nous avons

$$\frac{\beta}{\mu\sigma + \beta} \leq \lim_{t \rightarrow +\infty} J(t) \quad (\text{B.5.60})$$

Or, en procédant au changement de variable suivant dans $I_\beta(t)$,

$$u = x^\beta \text{ qui équivaut à } x = u^{\frac{1}{\beta}} \quad (\text{B.5.61})$$

qui donne

$$dx = \frac{1}{\beta} u^{\left(\frac{1}{\beta}-1\right)} du \quad (\text{B.5.62})$$

nous obtenons

$$\begin{aligned} I_\beta(t) &= \exp(-\mu\sigma - t^\beta) \int_1^t \exp(\mu\sigma x + x^\beta) dx \\ &= \exp(-\mu\sigma - t^\beta) \int_1^{t^\beta} \frac{1}{\beta} \exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right) u^{\left(\frac{1}{\beta}-1\right)} du \\ &= \frac{1}{\mu\sigma} \left\{ \exp(-\mu\sigma t - t^\beta) \right\} \left\{ \int_1^{t^\beta} \left(1 + \frac{1}{\beta} \mu\sigma u^{\left(\frac{1}{\beta}-1\right)}\right) \exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right) du - \right. \\ &\quad \left. \int_1^{t^\beta} \exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right) du \right\} \\ &= \frac{1}{\mu\sigma} \left\{ \exp(-\mu\sigma t - t^\beta) \right\} \left\{ \left[\exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right)\right]_1^{t^\beta} - \int_1^{t^\beta} \exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right) du \right\} \end{aligned}$$

$$\begin{aligned}
I_{\beta}(t) &= \frac{1}{\mu\sigma} \left\{ \exp(-\mu\sigma t - t^{\beta}) \right\} \left\{ \exp(\mu\sigma t + t^{\beta}) - \exp(\mu\sigma + 1) \right\} - \\
&\quad \frac{1}{\mu\sigma} \left\{ \exp(-\mu\sigma t - t^{\beta}) \right\} \int_1^{t^{\beta}} \exp\left(\mu\sigma u^{\frac{1}{\beta}} + u\right) du \\
&= \frac{1}{\mu\sigma} \left\{ 1 - \exp\left((\mu\sigma + 1) - (\mu\sigma t + t^{\beta})\right) - J(t) \right\}
\end{aligned} \tag{B.5.63}$$

Et comme

$$\lim_{t \rightarrow +\infty} \exp\left((\mu\sigma + 1) - (\mu\sigma t + t^{\beta})\right) = 0 \tag{B.5.64}$$

nous avons ainsi à la limite

$$\lim_{t \rightarrow +\infty} I_{\beta}(t) = \frac{1}{\mu\sigma} \left(1 - \lim_{t \rightarrow +\infty} J(t) \right) \tag{B.5.65}$$

Cette égalité B.5.65, ensemble avec l'inégalité B.5.60 donne alors

$$\lim_{t \rightarrow +\infty} I_{\beta}(t) \leq \frac{1}{\mu\sigma} \left(1 - \frac{\beta}{\mu\sigma + \beta} \right) \tag{B.5.66}$$

soit, en simplifiant le deuxième membre

$$\lim_{t \rightarrow +\infty} I_{\beta}(t) \leq \frac{1}{\mu\sigma + \beta} \tag{B.5.67}$$

On obtient ainsi, avec la définition de I_{β} qui est une fonction positive

$$0 \leq \lim_{t \rightarrow +\infty} I_{\beta}(t) \leq \frac{1}{\mu\sigma + \beta} \tag{B.5.68}$$

Et enfin, avec l'égalité B.5.31, nous avons

$$0 \leq \lim_{t \rightarrow +\infty} A_{\beta}^1(t) \leq \frac{\mu}{\mu + \frac{\beta}{\sigma}} \tag{B.5.69}$$

Les conclusions du théorème se déduisent alors de ces trois conclusions partielles (B.5.35 , B.5.52 et B.5.69) en utilisant le lemme B.5.2.3. \square

Le théorème suivant donne les variations de la fonction disponibilité dans le cas où la fiabilité suit une loi de Weibull et ce, suivant les valeurs du paramètre de forme β .

Théorème B.5.2.8. Variations de la disponibilité suivant le paramètre de forme

Soit σ , β et γ (avec $\sigma > 0$, $\beta > 0$ et $\gamma \in \mathbb{R}$) respectivement les paramètres d'échelle, de forme et de position de la loi de Weibull. Soit μ (avec $\mu > 0$) le taux de réparation de l'entité. Supposons que ce taux de réparation μ soit constant. Posons A_{β} la fonction disponibilité pour des valeurs de σ , γ , β et μ fixées. Alors :

a) Si $\beta \geq 1$, alors la fonction disponibilité A_{β} est une fonction décroissante, variant de 1 lorsque le temps $t = \gamma$ à une limite $A_{\beta}(\infty)$ lorsque le temps t tend vers plus l'infinie.

b) Si $0 < \beta < 1$ alors il existe un réel noté T_{β} (avec $T_{\beta} > \gamma$) tel que A_{β} soit décroissante sur l'intervalle $[\gamma, T_{\beta}]$ et croissante sur l'intervalle $[T_{\beta}, +\infty[$. Elle décroît alors de 1 lorsque le temps $t = \gamma$ à $A_{\beta}(T_{\beta})$ lorsque $t = T_{\beta}$, puis croît de $A_{\beta}(T_{\beta})$ lorsque $t = T_{\beta}$ à une limite $A_{\beta}(\infty)$ lorsque le

temps t tend vers plus l'infinie. Ce réel T_β vérifie l'équation suivante :

$$A_\beta(T_\beta) = \frac{\mu}{\mu + \beta \frac{(T_\beta - \gamma)^{(\beta-1)}}{\sigma^\beta}} \quad (\text{B.5.70})$$

Ces limites sont encadrées dans le théorème B.5.2.6.

Remarque B.5.2.9. En fait, comme il est rappelé plus haut, les cas où le paramètre de forme β est strictement entre 0 et 1 correspondent à la phase de jeunesse de l'entité. Les cas où $\beta = 1$ correspondent à la période de maturité de l'entité et les cas où β est strictement supérieur à 1 correspondent à sa phase de vieillesse. Ainsi, une observation à tirer du théorème B.5.2.8 est que si l'entité pouvait garder ses caractéristiques de jeunesse ($0 < \beta < 1$) indéfiniment, alors sa disponibilité croîtrait vers une limite de plus en plus proche de 1 (pour β décroissant vers 0) lorsque le temps t tend vers plus l'infinie. Cela signifie que ses performances augmenterait.

Preuve. Pour le cas où $\beta = 1$, comme c'est dit plus haut, la disponibilité suit la loi exponentielle et nous savons déjà qu'elle est décroissante.

Pour les autres cas, nous allons, en nous appuyant sur le lemme B.5.2.3, utiliser la fonction A_β^1 définie sur \mathbb{R}_+ par

$$A_\beta^1(t) = \left\{ \exp(-\mu\sigma t - t^\beta) \right\} \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx \right\} \quad (\text{B.5.71})$$

La fonction A_β^1 est dérivable sur \mathbb{R}_+^* (quelle que soit la valeur du paramètre de forme β) et nous avons

$$\begin{aligned} \frac{d}{dt} A_\beta^1(t) &= -(\mu\sigma + \beta t^{(\beta-1)}) \cdot \left\{ \exp(-\mu\sigma t - t^\beta) \right\} \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx \right\} \\ &\quad + \mu\sigma \exp(-\mu\sigma t - t^\beta) \cdot \exp(\mu\sigma t + t^\beta) \\ &= -(\mu\sigma + \beta t^{(\beta-1)}) \cdot \left\{ \exp(-\mu\sigma t - t^\beta) \right\} \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx \right\} + \mu\sigma \end{aligned} \quad (\text{B.5.72})$$

Pour étudier le sens de variation de A_β^1 , nous allons résoudre l'inéquation $\frac{d}{dt} A_\beta^1(t) > 0$. Cette inéquation est équivalente à

$$\exp(-\mu\sigma t - t^\beta) \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx \right\} < \frac{\mu\sigma}{\mu\sigma + \beta t^{(\beta-1)}} \quad (\text{B.5.73})$$

ou encore

$$\mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx < \frac{\mu\sigma}{\mu\sigma + \beta t^{(\beta-1)}} \exp(\mu\sigma t + t^\beta) - 1 \quad (\text{B.5.74})$$

soit

$$\int_0^t \exp(\mu\sigma x + x^\beta) dx < \frac{1}{\mu\sigma + \beta t^{(\beta-1)}} \exp(\mu\sigma t + t^\beta) - \frac{1}{\mu\sigma} \quad (\text{B.5.75})$$

Pour le cas où $\beta > 1$:

Dans ce cas, la fonction qui à un réel x associe $\frac{\exp(\mu\sigma x + x^\beta)}{\mu\sigma + \beta x^{(\beta-1)}}$ prend la valeur $\frac{1}{\mu\sigma}$ lorsque $x = 0$.

Ainsi, l'inéquation B.5.75 devient

$$\int_0^t \exp(\mu\sigma x + x^\beta) dx < \left[\frac{\exp(\mu\sigma t + t^\beta)}{\mu\sigma + \beta t^{(\beta-1)}} \right]_0^t \quad (\text{B.5.76})$$

Or la dérivée, définie sur \mathbb{R}_+^* , de la fonction qui à x associe $\frac{\exp(\mu\sigma t + t^\beta)}{\mu\sigma + \beta t^{(\beta-1)}}$ est

$$\frac{d}{dx} \left(\frac{\exp(\mu\sigma x + x^\beta)}{\mu\sigma + \beta x^{(\beta-1)}} \right) = \exp(\mu\sigma x + x^\beta) - \exp(\mu\sigma x + x^\beta) \cdot \left(\frac{\beta(\beta-1)x^{(\beta-2)}}{(\mu\sigma + \beta x^{(\beta-1)})^2} \right) \quad (\text{B.5.77})$$

Donc l'inéquation B.5.76 revient à

$$\int_0^t \exp(\mu\sigma x + x^\beta) dx < \int_0^t \exp(\mu\sigma x + x^\beta) dx - \int_0^t \exp(\mu\sigma x + x^\beta) \cdot \left(\frac{\beta(\beta-1)x^{(\beta-2)}}{(\mu\sigma + \beta x^{(\beta-1)})^2} \right) dx \quad (\text{B.5.78})$$

soit

$$0 < - \int_0^t \exp(\mu\sigma x + x^\beta) \cdot \left(\frac{\beta(\beta-1)x^{(\beta-2)}}{(\mu\sigma + \beta x^{(\beta-1)})^2} \right) dx \quad (\text{B.5.79})$$

Et comme $\beta > 1$, alors, nous avons

$$\int_0^t \exp(\mu\sigma x + x^\beta) \cdot \left(\frac{\beta(\beta-1)x^{(\beta-2)}}{(\mu\sigma x + \beta x^{(\beta-1)})^2} \right) dx \geq 0 \quad (\text{B.5.80})$$

Ainsi, dans le cas où $\beta > 1$, il est impossible d'avoir $\frac{d}{dt} A_1^\beta(t) > 0$. La fonction A_1^β est donc non croissante sur \mathbb{R}_+ . Par l'intermédiaire du lemme B.5.2.3, nous concluons donc que la fonction A_β est non croissante sur $[\gamma, +\infty[$, pour $\beta > 1$.

Pour le cas où $0 < \beta < 1$:

Reprenons l'inéquation B.5.75 avec une inégalité large

$$\int_0^t \exp(\mu\sigma x + x^\beta) dx \leq \frac{1}{\mu\sigma + \beta t^{(\beta-1)}} \exp(\mu\sigma t + t^\beta) - \frac{1}{\mu\sigma} \quad (\text{B.5.81})$$

et posons pour $t > 0$,

$$g(t) = \int_0^t \exp(\mu\sigma x + x^\beta) dx - \frac{1}{\mu\sigma + \beta t^{(\beta-1)}} \exp(\mu\sigma t + t^\beta) \quad (\text{B.5.82})$$

L'inéquation B.5.81 s'écrit alors

$$g(t) \leq -\frac{1}{\mu\sigma} \quad (\text{B.5.83})$$

La fonction g est dérivable sur \mathbb{R}_+^* et sa dérivée est donnée par

$$\begin{aligned} \frac{d}{dt} g(t) &= \exp(\mu\sigma t + t^\beta) - \exp(\mu\sigma t + t^\beta) + \exp(\mu\sigma t + t^\beta) \cdot \frac{\beta(\beta-1)t^{(\beta-2)}}{(\mu\sigma + \beta t^{(\beta-1)})^2} \\ &= \frac{\beta(\beta-1)t^{(\beta-2)}}{(\mu\sigma + \beta t^{(\beta-1)})^2} \exp(\mu\sigma t + t^\beta) \end{aligned} \quad (\text{B.5.84})$$

Comme $0 < \beta < 1$, la dérivée $\frac{d}{dt}g$ de g est alors négative compte tenu de l'égalité B.5.84. La fonction g est donc décroissante. En plus, $g(0) = 0$ (par prolongement par continuité).

Par ailleurs, $g(t)$ tend vers moins l'infinie lorsque t tend vers plus l'infinie.

En effet :

$$g(t) = \int_0^t \frac{d}{dx}g(x)dx \quad (\text{B.5.85})$$

l'expression de $\frac{d}{dx}g(x)$ étant définie dans l'équation B.5.84. Or, d'après les propriétés des fonctions exponentielles, pour toute fonction polynôme P , il existe un réel X tel que pour tout x plus grand que X , $P(x) < \exp(\mu\sigma x + x^\beta)$. Ainsi, il existe un réel X tel que pour tout x plus grand que X , l'on ait

$$x^2 < \exp(\mu\sigma x + x^\beta) \quad (\text{B.5.86})$$

Et alors pour tout x plus grand que X , nous aurons

$$\frac{\beta(\beta-1)x^{(\beta-2)}x^2}{(\mu\sigma + \beta x^{(\beta-1)})^2} > \frac{\beta(\beta-1)x^{(\beta-2)}}{(\mu\sigma + \beta x^{(\beta-1)})^2} \exp(\mu\sigma x + x^\beta) \quad (\text{B.5.87})$$

c'est à dire

$$\frac{\beta(\beta-1)x^\beta}{(\mu\sigma + \beta x^{(\beta-1)})^2} > \frac{d}{dx}g(x) \quad (\text{B.5.88})$$

Or

$$\lim_{x \rightarrow +\infty} \left(\frac{\beta(\beta-1)x^\beta}{(\mu\sigma + \beta x^{(\beta-1)})^2} \right) = -\infty \quad (\text{B.5.89})$$

Donc

$$\lim_{x \rightarrow +\infty} \frac{d}{dx}g(x) = -\infty \quad (\text{B.5.90})$$

La fonction $\frac{d}{dx}g$ n'est donc pas intégrable sur $[0, +\infty[$. Par conséquent (et puisque $\frac{d}{dx}g(x)$ est une fonction négative) nous avons,

$$\lim_{x \rightarrow +\infty} g(t) = \lim_{x \rightarrow +\infty} \int_0^t g'(x)dx = -\infty \quad (\text{B.5.91})$$

En résumé, la fonction g est strictement décroissante, prend la valeur 0 en 0 et tend vers moins l'infinie lorsque t tend vers plus l'infinie. Il existe donc un unique réel t_β dans \mathbb{R}_+^* vérifiant la condition

$$\begin{aligned} g(t_\beta) &= -\frac{1}{\mu\sigma} \\ g(t) &> -\frac{1}{\mu\sigma} \quad \forall t, 0 < t < t_\beta \\ g(t) &< -\frac{1}{\mu\sigma} \quad \forall t, t > t_\beta \end{aligned} \quad (\text{B.5.92})$$

La fonction A_1^β est donc strictement décroissante sur $[0, t_\beta]$ et strictement croissante sur $[t_\beta, +\infty[$.

Ainsi, d'après le lemme B.5.2.3, la fonction disponibilité A_β est strictement décroissante sur $[\gamma, T_\beta]$ et strictement croissante sur $[T_\beta, +\infty]$ où $T_\beta = \sigma t_\beta + \gamma$.

En outre, la dérivée de la fonction disponibilité est donnée par

$$\frac{d}{dt}A_\beta(t) = - \left(\mu + \frac{\beta}{\sigma} \left(\frac{t-\gamma}{\sigma} \right)^{(\beta-1)} \right) A_\beta(t) + \mu \quad (\text{B.5.93})$$

Et le fait que $\frac{d}{dt}A_\beta(T_\beta) = 0$ donne

$$A_\beta(T_\beta) = \frac{\mu}{\mu + \beta \frac{(T_\beta - \gamma)^{(\beta-1)}}{\sigma^\beta}} \quad (\text{B.5.94})$$

□

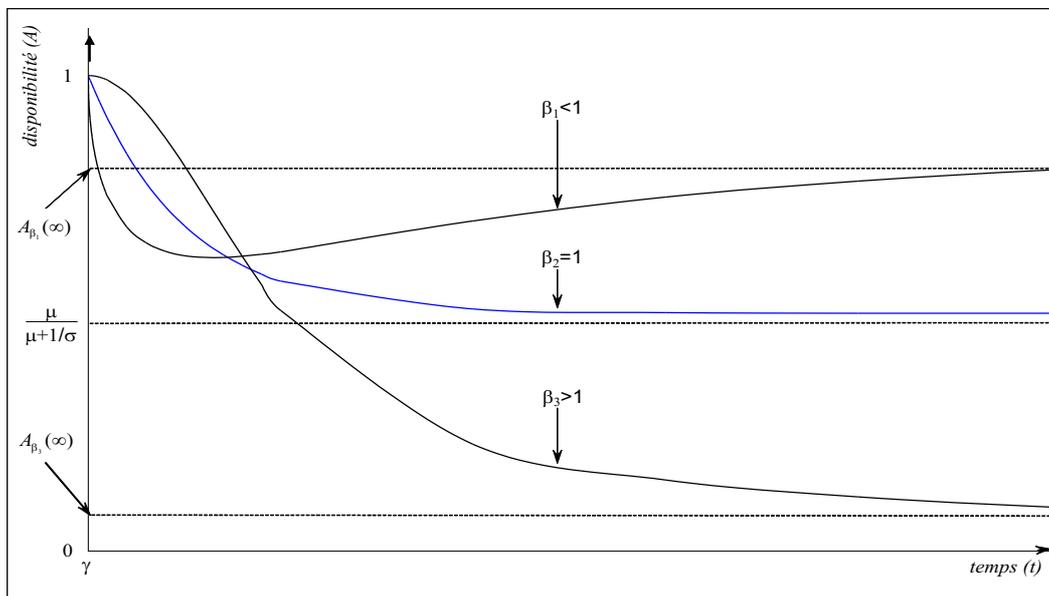


Figure B.2 – Évolution de la disponibilité suivant les valeurs du paramètre de forme

En s'aidant de ce théorème, il est aisé de déduire l'existence de la limite lorsque le temps t tend vers plus l'infinie de la fonction disponibilité. Cela est dans le corollaire suivant. Comme nous le disions un peu plus tôt, c'est le contraire qui serait intuitivement surprenant.

Corollaire B.5.2.10. *Pour toutes les valeurs du paramètre de forme β , la fonction disponibilité admet une limite notée $A_\beta(\infty)$ lorsque le tend t tend vers plus l'infinie.*

Preuve. Lorsque $\beta \geq 1$, la fonction disponibilité A_β est décroissante d'après le théorème B.5.2.8. Or A_β est une fonction positive, donc minorée par 0. Elle admet donc une limite lorsque le temps t tend vers plus l'infinie.

Lorsque $0 < \beta < 1$, le théorème B.5.2.8 nous dit que la fonction disponibilité A_β est croissante à partir d'une valeur notée T_β du temps t . Or A_β est une fonction qui a ses valeurs dans l'intervalle $[0, 1]$. Elle est donc majorée par 1. Ainsi, elle admet une limite lorsque le temps t tend vers plus l'infinie. □

Cette limite, on sait la calculer explicitement lorsque $\beta = 1$ et lorsque $0 < \beta < 1$ ou $\beta > 1$, on sait l'encadrer avec une bonne précision dans le théorème B.5.2.6.

Enfin, le théorème suivant donne les fonctions limites lorsque le paramètre de forme β tend vers 0 et vers plus l'infinie de la fonction disponibilité A_β .

Théorème B.5.2.11. Fonctions limites de la fonction disponibilité

Soit σ , β et γ ($\sigma > 0$, $\beta > 0$ et $\gamma \in \mathbb{R}$) respectivement les paramètres d'échelle, de forme et de position de la loi de Weibull. Soit μ ($\mu > 0$) le taux de réparation de l'entité. Supposons que ce taux de réparation μ soit constant. Posons A_β la fonction disponibilité pour des valeurs de σ , γ , β et μ fixées.

Alors A_β admet une fonction limite notée A_0 (resp. A_∞) lorsque le paramètre de forme β tend vers 0 (resp. plus l'infinie). Ces fonctions limites sont définies par :

$$A_0 : [\max(0, \gamma), +\infty[\longrightarrow [0, 1]$$

$$t \longmapsto A_0(t) = \begin{cases} 1 & \text{si } t = \gamma \\ 1 - (1 - e^{-1}) \exp(-\mu(t - \gamma)) & \text{si } t > \gamma \end{cases} \quad (\text{B.5.95})$$

$$A_\infty : [\max(0, \gamma), +\infty[\longrightarrow [0, 1]$$

$$t \longmapsto A_\infty(t) = \begin{cases} 1 & \text{si } \gamma \leq t < \gamma + \sigma \\ e^{-1} & \text{si } t = \gamma + \sigma \\ 0 & \text{si } t > \gamma + \sigma \end{cases} \quad (\text{B.5.96})$$

Ces deux fonctions limites sont représentées sur la figure B.3.

Preuve. En utilisant l'égalité B.5.12 dans la preuve du lemme B.5.2.3, nous allons encore une fois baser la démonstration de ce théorème sur la fonction A_β^1 définie sur \mathbb{R}_+ par

$$A_\beta^1(t) = \left\{ \exp\left(-[\mu\sigma t + t^\beta]\right) \right\} \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^\beta) dx \right\} \quad (\text{B.5.97})$$

Pour le cas où $\beta > 0$ tend vers 0 :

Pour β fixée, $A_\beta^1(0) = 1$. Donc $\lim_{\beta \rightarrow 0} A_\beta^1(0) = 1$.

Soit t un réel fixé ($t > 0$). Nous avons

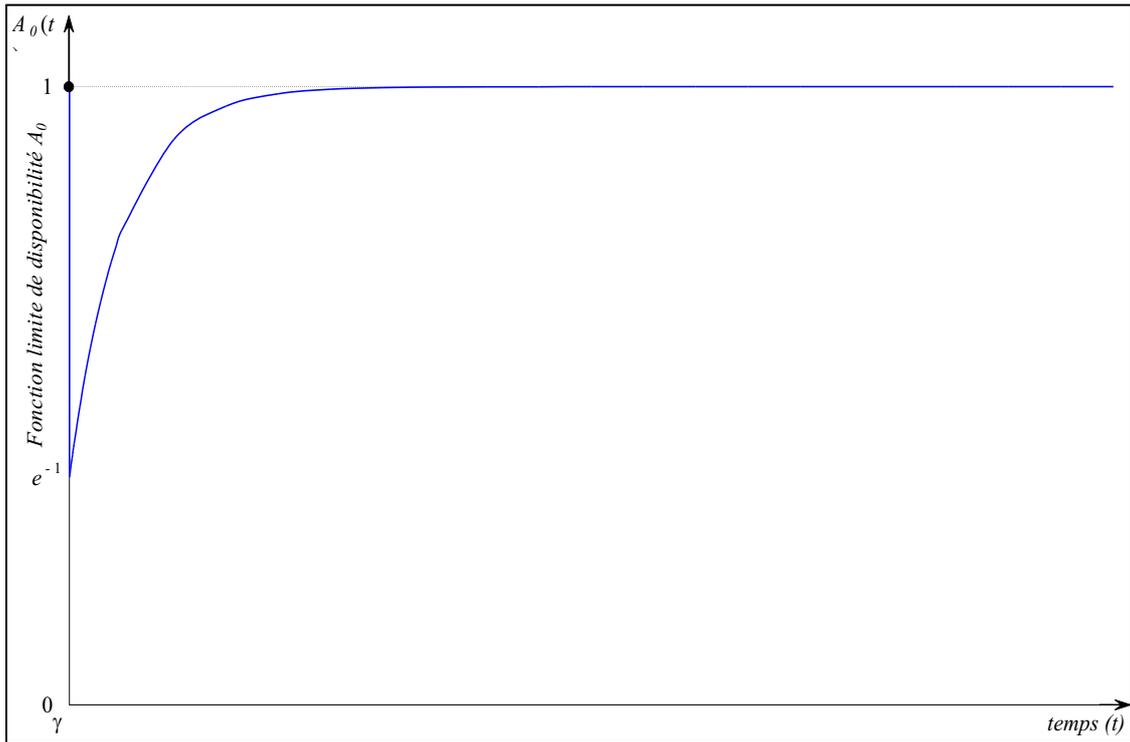
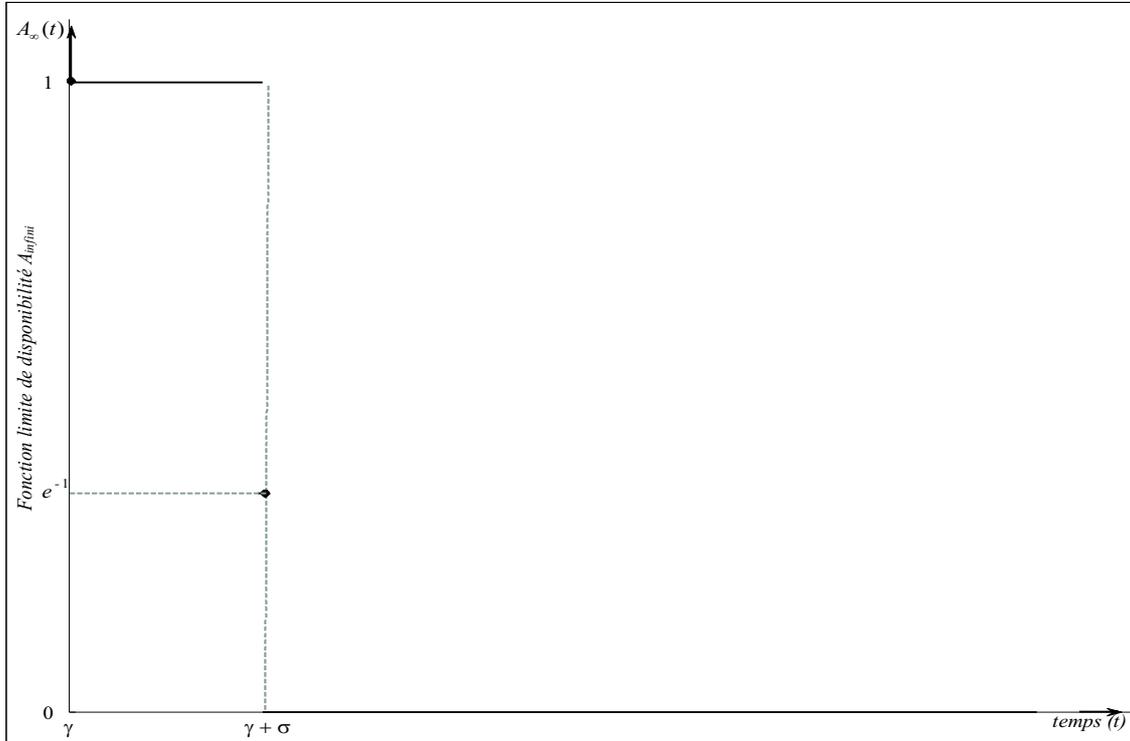
$$\int_0^t \exp(\mu\sigma x + x^\beta) dx = \int_{]0, t]} \exp(\mu\sigma x + x^\beta) dx \quad (\text{B.5.98})$$

Or, pour tout réel x strictement positif, on a

$$\lim_{\beta \rightarrow 0} x^\beta = 1 \quad (\text{B.5.99})$$

Donc

$$\begin{aligned} \lim_{\beta \rightarrow 0} A_\beta^1(t) &= \exp(-[\mu\sigma t + 1]) \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + 1) dx \right\} \\ &= \exp(-\mu\sigma t - 1) \cdot \left\{ 1 + [\exp(\mu\sigma x + 1)]_0^t \right\} \\ &= \exp(-\mu\sigma t - 1) \cdot (1 + \exp(\mu\sigma t + 1) - e) \\ &= \exp(-\mu\sigma t - 1) + 1 - \exp(-\mu\sigma t) \end{aligned}$$

(a) Fonction disponibilité limite A_0 (b) Fonction disponibilité limite A_∞ Figure B.3 – Fonctions disponibilité limites A_0 et A_∞

soit

$$\lim_{\beta \rightarrow 0} A_{\beta}^1(t) = 1 - (1 - e^{-1}) \exp(-\mu\sigma t) \quad (\text{B.5.100})$$

En résumé,

$$A_0^1(t) = \begin{cases} 1 & \text{si } t = 0 \\ 1 - (1 - e^{-1}) \exp(-\mu\sigma t) & \text{si } t > 0 \end{cases} \quad (\text{B.5.101})$$

Or, d'après l'égalité B.5.12, nous avons $A(t) = A^1\left(\frac{t-\gamma}{\sigma}\right)$. D'où

$$A_0(t) = \begin{cases} 1 & \text{si } t = \gamma \\ 1 - (1 - e^{-1}) \exp(-\mu(t - \gamma)) & \text{si } t > \gamma \end{cases} \quad (\text{B.5.102})$$

Pour le cas où β tend vers plus l'infinie :

Tout comme dans le cas où β tend vers 0, nous avons $\lim_{\beta \rightarrow +\infty} A_{\beta}^1(0) = 1$.

Soit t un réel fixé ($t > 0$). Nous avons

$$\begin{aligned} A_{\beta}^1(t) &= \exp(-\mu\sigma t - t^{\beta}) \cdot \left\{ 1 + \mu\sigma \int_0^t \exp(\mu\sigma x + x^{\beta}) dx \right\} \\ &= \exp(-\mu\sigma t - t^{\beta}) \cdot \left\{ 1 + \mu\sigma \int_{]0,t[} \exp(\mu\sigma x + x^{\beta}) dx \right\} \\ &= \exp(-\mu\sigma t - t^{\beta}) + \mu\sigma \int_{]0,t[} \exp(-\mu\sigma(t-x) - (t^{\beta} - x^{\beta})) dx \\ &= \exp(-\mu\sigma t + t^{\beta}) + \mu\sigma \int_{]0,t[} \exp\left(-\mu\sigma(t-x) - t^{\beta} \left(1 - \left(\frac{x}{t}\right)^{\beta}\right)\right) dx \end{aligned} \quad (\text{B.5.103})$$

Or, pour $t > 0$ fixé et pour tout réel x tel que $0 < x < t$, nous avons $0 < \frac{x}{t} < 1$. Ainsi,

$$\lim_{\beta \rightarrow +\infty} \left(\frac{x}{t}\right)^{\beta} = 0 \quad (\text{B.5.104})$$

Donc lorsque β tend vers plus l'infinie, $A_{\beta}^1(t)$ est équivalente à $I(t)$ avec

$$\begin{aligned} I(t) &= \exp(-\mu\sigma t - t^{\beta}) + \mu\sigma \int_{]0,t[} \exp(-\mu\sigma(t-x) - t^{\beta}) dx \\ &= \exp(-\mu\sigma t - t^{\beta}) + \exp(-t^{\beta}) \cdot [\exp(\mu\sigma(x-t))]_0^t \\ &= \exp(-\mu\sigma t - t^{\beta}) + \exp(-t^{\beta}) \{1 - \exp(-\mu\sigma t)\} \\ &= \exp(-t^{\beta}) \end{aligned} \quad (\text{B.5.105})$$

Ainsi :

Si $t = 1$, alors $\lim_{\beta \rightarrow +\infty} A_{\beta}^1(t) = e^{-1}$.

Si $0 < t < 1$, alors $\lim_{\beta \rightarrow +\infty} (t^{\beta}) = 0$ et donc $\lim_{\beta \rightarrow +\infty} A_{\beta}^1(t) = 1$.

Si $t > 1$, alors $\lim_{\beta \rightarrow +\infty} (t^{\beta}) = +\infty$ et donc $\lim_{\beta \rightarrow +\infty} A_{\beta}^1(t) = 0$.

En résumé, nous avons

$$A_{\infty}^1(t) = \begin{cases} 1 & \text{si } 0 \leq t < 1 \\ e^{-1} & \text{si } t = 1 \\ 0 & \text{si } t > 1 \end{cases} \quad (\text{B.5.106})$$

Or, d'après l'égalité B.5.12, nous avons $A(t) = A^1\left(\frac{t-\gamma}{\sigma}\right)$. D'où

$$A_{\infty}(t) = \begin{cases} 1 & \text{si } \gamma \leq t < \gamma + \sigma \\ e^{-1} & \text{si } t = \gamma + \sigma \\ 0 & \text{si } t > \gamma + \sigma \end{cases} \quad (\text{B.5.107})$$

□

B.6 Conclusion

Dans ce chapitre, nous avons analysé la fonction disponibilité d'un composant dont la loi de fiabilité n'est pas exponentielle. Ce cas de la loi exponentielle est très largement utilisé dans la littérature pour des raisons qui sont spécifiquement liées à sa structure. C'est une loi assez simple dans son utilisation et les manipulations mathématiques sont assez simples. Nous avons montré que même lorsque la loi de fiabilité n'est pas exponentielle mais que les taux de défaillance et de réparation du composant peuvent être exprimés analytiquement en fonction du temps, il est possible de déterminer sa fonction disponibilité. Nous avons ensuite considéré le cas où la loi de fiabilité du composant est de Weibull (où la loi exponentielle est alors un cas particulier). Nous avons analysé de façon exhaustive le comportement de la fonction disponibilité dans un cas où le taux de réparation est quelconque mais analytique. Nous avons ensuite étudié le cas où ce taux de réparation est constant. Les différents résultats permettent de pouvoir utiliser cette fonction disponibilité dans des cas beaucoup plus complexes que celui de la loi exponentielle.

Bibliographie

- [1] Adjallah, K.H., **K.P., Adzakpa**, A., Ivanov, C., Varnier, and N., Zerhouni. Approach to remote maintenance management of distributed industrial complexes. In *Proceedings of MIM2002, 5th International Conference on Managing Innovations in Manufacturing*, p. 101–109, Milwaukee, Wisconsin, USA, September 9-11, 2002.
- [2] **Adzakpa, K.P.** and K. H., Adjallah. Minimisation du temps de séjour avec dates d'arrivée inégales des tâches et set-ups : Application à la maintenance. In *Actes du 5e Congrès International Pluridisciplinaire : Qualité et Sécurité de Fonctionnement (QUALITA'2003)*, p. 179–186, Nancy, France, 18-20 Mars, 2003.
- [3] **Adzakpa, K.P.** and K. H., Adjallah. Minimisation du temps de séjour pondéré avec dates de disponibilité inégales des tâches : Application à la maintenance. In *Actes de la 4e Conférence Francophone de Modélisation et Simulation (MOSIM'03)*, p. 635–642, Toulouse, France, 23-25 Avril 2003.
- [4] **Adzakpa, K.P.** and K. H., Adjallah. Minimizing maintenance cost involving flow-time and tardiness penalty with unequal release dates. In *Proceedings of IMS'2003, International Conference on Intelligent Maintenance Systems*, p. 787–797, Xi'an, China, October 30 - November 1 2003. National University of Defense Technology Press.
- [5] **Adzakpa, K.P.** and K. H., Adjallah. Minimizing the total weighted flow-time of jobs with set-up times and unequal release dates: Application to minimizing maintenance operations cost. In *Proceedings CD-rom of CESA'2003 IMACS Multiconference, Computational Engineering in Systems Applications*, p. S3-R-00-0046, Lille, France, July 9-11, 2003.
- [6] **Adzakpa, K.P.** and K. H., Adjallah. Minimizing maintenance cost involving flow-time and tardiness penalty with unequal release dates and set-up times. In *Proceedings CD-rom of MARCON 2004, MAintenance and Reliability CONference*, Knoxville, TN, USA, May 2-5 2004.
- [7] **Adzakpa, K.P.** and K. H., Adjallah. Monte-carlo simulation and flow-time minimization with set-up times and unequal release dates for maintenance planning. In *Proceedings CD-rom of IMS'2004, International Conference on Intelligent Maintenance Systems*, Arles, France, July 15-17 2004.
- [8] **Adzakpa, K.P.**, K. H., Adjallah, and F., Yalaoui. Un algorithme on-line d'allocation de ressources aux systèmes répartis par optimisation multicritère. In *Actes de ROADEF 2002 : 4e Congrès de la société française de recherche opérationnelle et d'aide à la décision*, p. 25–26, Télécom Paris, Paris, France, 20-22 Février, 2002.

- [9] **Adzakpa, K.P.**, K.H., Adjallah, and J., Lee. A new effective heuristic for the intelligent management of the preventive maintenance of the distributed systems. *Advanced Engineering Informatics*, 17 (3-4), p. 151–163, 2003.
- [10] **Adzakpa, K.P.**, K.H., Adjallah, and F., Yalaoui. On-line maintenance job scheduling and assignment to resources in distributed systems by heuristic-based optimization. *Journal of Intelligent Manufacturing*, 15 (2), p. 131–140, 2004.
- [11] Alfredsson, P. Optimization of multi-echelon repairable item inventory systems with simultaneous location of repair facilities. *European Journal of Operational Research*, 99, p. 584–595, 1997.
- [12] Anily, S., C.A., Glass, and R., Hassin. The scheduling of maintenance service. *Discrete Applied Mathematics*, 82, p. 27–42, 1998.
- [13] Archibald, T.W. and R., Dekker. Modified block replacement for multi-component systems. *IEEE Transactions on Reliability*, 45 (1), p. 75–83, 1996.
- [14] Assaf, D. and J.G., Shantikumar. Optimal group maintenance policies with continuous and periodic inspections. *Management Science*, 33, p. 1440–1452, 1987.
- [15] Aven, T. Condition based replacement - a counting process approach. *Reliability Engineering and System Safety*, 51, p. 275–291, 1996.
- [16] Aven, T. and R., Dekker. A useful framework for optimal replacement models. *Reliability Engineering and System Safety*, 51, p. 61–67, 1997.
- [17] Ayyub, B.M. and R.H., McCuen. *Probability, Statistics, and Reliability for Engineers and Scientists*. Chapman and Hall/CRC, 2003.
- [18] Bagai, I. and K., Jain. Improvement, deterioration and optimal replacement under age-replacement with minimal repair. *IEEE Transaction on Reliability*, 43, p. 156–162, 1994.
- [19] Baker, R.D. and A.H., Christer. Review of OR modelling of engineering aspects of maintenance. *European Journal of Operational Research*, 73, p. 407–422, 1994.
- [20] Barbera, F., H., Schneider, and P., Kelle. A condition based model with exponential failures and fixed inspection intervals. *Journal of the Operational research Society*, 47 (8), p. 1037–1045, 1996.
- [21] Belouadah, H., M.E., Posner, and C.N., Potts. Scheduling with release dates on a single machine to minimize total weighted completion time. *Discrete Applied Mathematics*, 36, p. 213–231, 1992.
- [22] Bennour, M., C., Bloch, and N., Zerhouni. Modélisation intégrée de maintenance et de production. In *MOSIM' 01, Actes de la Troisième Conférence Francophone de Modélisation et Simulation*, p. 805–810, Troyes, France, Avril 25-27, 2001.
- [23] Berg, M.P. The marginal cost analysis and its application to repair and replacement policies. *European Journal of Operational Research*, 82, p. 214–224, 1995.
- [24] Bianco, L. and S., Ricciardelli. Scheduling of a single machine to minimize total weighted completion time subject to release dates. *Naval Research Logistics Quarterly*, 29, p. 151–167, 1982.

- [25] Block, H.W., W.S., Borges, and T.H., Savit. Age-dependent minimal repair. *Journal of Applied Probability*, 22, p. 370–385, 1985.
- [26] Block, H.W., N.A., Langberg, and T.H., Savit. Repair replace policies. *Journal of Applied Probability*, 27, p. 194–206, 1993.
- [27] Boland, J.P. Periodic replacement when minimal repair costs vary with time. *Naval Research Logistics Quarterly*, 29, p. 541–546, 1982.
- [28] Boland, J.P. and F., Poschan. Periodic replacement with increasing minimal repair cost at failure. *Operations Research*, 30, p. 1183–1189, 1982.
- [29] Bon, J.-L. *Fiabilité des systèmes : Méthodes mathématiques*. Masson, Paris, 1995.
- [30] Bouche, M., V., Plauchu, and D., Retour. *Systèmes Experts en Maintenance*, chapitre Les Systèmes Experts Appliquées à la Miantenance Modifient-ils la Donne?, p. 102–112. Masson, Paris, 1987.
- [31] Boucly, F. *Le Management de la Maintenance Assisté par Ordinateur*. AFNOR, Paris, 1990.
- [32] Boucly, F. and A., Ogus. *Management de la Maintenance*. AFNOR, Paris, 1987.
- [33] Boulenger, A. *Vers le Zero Panne avec la Maintenance Conditionnelle*. AFNOR, Paris, 1988.
- [34] Castanier, B. *Modélisation Stochastique et Optimisation de la Maintenance Conditionnelle des Systèmes à Dégradation Graduelle*. PhD thesis, Université de Technologie de Troyes, Troyes, France, 2001.
- [35] Chakravarty, A.K., J.B., Orlin, and U.G., Rothblum. A partitioning problem with additive objective with an application to optimal inventory grouping for joint replenishment. *Operations Research*, 30, p. 1018–1022, 1982.
- [36] Chauvet, F. and J.-M., Proth. Scheduling and controlling work-in-process: An on-line study for shop problems. Research Report N3950, INRIA-Lorraine, Metz, France, May 2000.
- [37] Chen, M. and R.M., Feldman. Optimal replacement policies with minimal repair and age-dependent costs. *European Journal of Operational Research*, 98, p. 75–84, 1997.
- [38] Cho, D.I. and M., Parlar. A survey of maintenance models for multi-unit systems. *European Journal of Operational Research*, 51, p. 1–23, 1991.
- [39] Christer, A.H. Delay time models of industrial maintenance problems. *Journal of Operational Research Society*, 35, p. 401–406, 1984.
- [40] Christer, A.H. and W., Wang. A model of condition monitoring of a production plant. *International Journal of Production Research*, 30 (9), p. 2199–2211, 1992.
- [41] Chu, C. One-machine scheduling for minimizing total flow time with release dates. In *Proceedings of Rensselaer's Second Conference on C.I.M.*, p. 570–576, May 21-23, 1990.
- [42] Chu, C. A branch-and-bound algorithm to minimize total flow time with unequal release dates. *Naval Research Logistics*, 39, p. 859–875, 1992.
- [43] Chu, C. A branch-and-bound algorithm to minimize total tardiness with different release dates. *Naval Research Logistics*, 39, p. 265–283, 1992.

- [44] Chu, C. Efficient heuristics to minimize total flow time with release dates. *Operations Research Letters*, 12, p. 321–330, 1992.
- [45] Chu, C. and M.-C., Portmann. Some new efficient methods to solve the $n/1/r_i/\sum t_i$ scheduling problem. *European Journal of Operational Research*, 58, p. 404–413, 1992.
- [46] Cléroux, R., S., Dubuc, and C., Tilquin. The age replacement problem with minimal repair and random repair cost. *Operations Research*, 27, p. 1158–1167, 1979.
- [47] Daffua, S.O., A., Raouf, and J.D., Campbell. *Planning and Control of Maintenance Systems: Modelling and Analysis*. John Wiley, New York, NY, 1999.
- [48] Dekker, R. Integrating optimisation, priority setting, planning and combining of maintenance activities. *European Journal of Operational Research*, 82, p. 225–240, 1995.
- [49] Dekker, R. Applications of maintenance optimization models: a review and analysis. *Reliability Engineering and System Safety*, 51, p. 229–240, 1996.
- [50] Dekker, R. and M.C., Dijkstra. Opportunity-based age replacement: exponentially distributed times between opportunities. *Naval research Logistics*, 39, p. 175–190, 1992.
- [51] Dekker, R. and I.F.K., Roelvink. Marginal cost criteria for preventive replacement of a group of components. *European Journal of Operational Research*, 84, p. 467–480, 1995.
- [52] Dekker, R. and P.A., Scarf. On the impact of optimization models in maintenance decision making: The state of the art. *Reliability Engineering and Systems Safety*, 60 (2), p. 111–119, 1998.
- [53] Dekker, R. and E., Smeitink. Opportunity-based block replacement: The single component case. *European Journal of Operational Research*, 53, p. 46–63, 1991.
- [54] Dekker, R. and E., Smeitink. Preventive maintenance at opportunity of restricted duration. *Naval research Logistics*, 39, p. 335–353, 1994.
- [55] Dekker, R., A.C.J.M., Smit, and J.A., Losekoot. Combining maintenance activities in an operational planning phase: A set-partitioning approach. *IMA Journal of Mathematics Applied in Business and Industry*, 3, p. 315–331, 1992.
- [56] Dekker, R., F.A., van der Duyn Schouten, and R., Wilderman. A review of multi-component maintenance models with economic dependence. *Mathematical Models of Operations Research*, 45, p. 411–435, 1997.
- [57] Dekker, R., R.E., Wildeman, and F.A., van der Duyn Schouten. Review of multi-component models with economic dependence. *Zor/Mathematical Methods of Operations Research*, 45, p. 411–435, 1997.
- [58] Derman, C., G.J., Lieberman, and S.M., Ross. On the optimal assignment of servers and a repairman. *Journal of Applied Probability*, 17, p. 703–715, 1980.
- [59] Dorf, R.C., editor. *Technology Management Handbook*, chapitre Reliability and Maintainability. Boca Raton: CRC Press, 2000.
- [60] Dubi, A. *Monte Carlo Applications in Systems Engineering*. J. Wiley and Sons, c2000.
- [61] Duron, C. *Ordonnancement en Temps-Réel des Activités des Radars*. PhD thesis, Université de Metz - UFR MIM, France, 2002.

- [62] Feldman, R.M. Optimal replacement with semi-markov shock models. *Journal of Applied Probability*, 13, p. 108–117, 1976.
- [63] Fishman, G. S. *Monte Carlo: Concepts, Algorithms and Applications*. Springer, New-York, c1996.
- [64] Fox, B. L. *Strategies for Quasi-Monte Carlo*. Kluwer Academic Publishers, Boston, 1999.
- [65] Francastel, J.C. *La Fonction Maintenance*. AFNOR, Paris, 1999.
- [66] Francastel, J.C. *Ingenieurie de la Maintenance*. Dunod, Paris, 2003.
- [67] Frenk, J.B.G., R., Dekker, and M.J., Klein. A note on the marginal cost approach in maintenance. *Journal of Optimization Theory and Applications*, 94 (3), p. 771–781, 1997.
- [68] Frostig, E. Optimal policies for machine repairman problems. *Journal of Applied Probability*, 30, p. 703–715, 1993.
- [69] Frostig, E. Optimal allocation of machine repairman in order to maximize some reward functions. *Probability in the Engineering and Informational Services*, 9, p. 633–645, 1995.
- [70] Frostig, E. Jointly optimal allocation of a repairman and optimal control of service rate for machine repairman problem. *European Journal of Operational Research*, 116, p. 274–280, 1999.
- [71] Fry, T.D., R.D., Amstrong, and I.D., Rosen. Single machine scheduling to minimize mean absolute lateness: A heuristic solution. *Computers and Operations Research*, 17 (1), p. 105 – 112, 1990.
- [72] Gentle, J. E. *Random Number Generation and Monte Carlo Methods*. Springer, New York, c1998.
- [73] Gits, C.W. On the maintenance concepts for a technical system: II. literature review. *Maintenance Management International*, 6, p. 181–196, 1986.
- [74] Graves, G.H. and C.Y., Lee. Scheduling maintenance and semi resumable jobs on a single machine. *Naval Research Logistics*, 46, p. 845–863, 1999.
- [75] Hariri, A.M.A. and C.N., Potts. An algorithm for single machine sequencing with release dates to minimize total weighted completion time. *Discrete Applied Mathematics*, 5, p. 99–109, 1983.
- [76] Haurie, A. Time scale decomposition in production planning for unreliable flexible manufacturing systems. *European Journal of Operational Research*, 82, p. 339–358, 1995.
- [77] Haurie, A. and P., L'Ecuyer. A stochastic control approach to group preventive replacement in a multicomponent system. *IEEE Transactions on Automatic Control*, 27 (2), p. 387–393, 1982.
- [78] Haurie, A. and Ch., van Delft. Turnpikes in flow control models of unreliable manufacturing systems. *European Journal of Operational Research*, 82, p. 359–372, 1995.
- [79] Heidergott, B. Optimisation of a single-component maintenance system: A smoothed perturbation analysis approach. *European Journal of Operational Research*, 119, p. 181–190, 1999.
- [80] Hellekalek, P. and G., Larcher. *Random and Quasi-Random Point Sets*. Springer, New-York, c1998.

- [81] Héng, J. *Pratique de la Maintenance Préventive*. Dunod, 2002.
- [82] Hosseini, M.M., R.M., Kerr, and R.B., Randall. A hybrid maintenance model with imperfect inspection for a system with deterioration and poisson failure. *Journal of Operations Research Society*, 50, p. 1229–1243, 1999.
- [83] Hosseini, M.M., R.M., Kerr, and R.B., Randall. An inspection model with minimal and major maintenance for a system with deterioration and poisson failure. *IEEE Transactions on Reliability*, 49 (1), p. 88–98, 2000.
- [84] Hwang, H.-S. Inventory models for both deteriorating and ameliorating items. *Computers and Industrial Engineering*, 37, p. 257–260, 1999.
- [85] Iskandar, B.P. and H., Sandoh. An extended opportunity-based age replacement policy. *RAIRO Operations Research*, 34, p. 145–154, 2000.
- [86] Jardine, A.K.S. and J.A., Buzacott. Equipment reliability and maintenance. *European Journal of Operational Research*, 19, p. 285–296, 1985.
- [87] Kaio, N. and S., Osaki. Inspection policies: comparison and modification. *Operations Research*, 22 (4), p. 387–400, 1988.
- [88] Kao, E.P.C. Optimal replacement rules when changes of state are semi-markovian. *Operations Research*, 21, p. 1231–1249, 1973.
- [89] Katehakis, N.N. and C., Derman. Optimal repair allocation in a series system. *Mathematics and Operation Research*, 9, p. 615–623, 1984.
- [90] Katehakis, N.N. and C., Melolidakis. Dynamic repair allocation for a k-out-of-n system maintained by distinguishable repairmen. *Probability in the Engineering and the Informational Sciences*, 2, p. 51–62, 1988.
- [91] Kelly, A. *Maintenance Planning and Control*. Butterworths, Oxford, 1984.
- [92] Kelly, A. *Maintenance Organization and Systems*. Butterworth-Heinemann, Oxford, 1997.
- [93] Knezevic, J. *Systems Maintainability*. Chapman and Hall, London, 1997.
- [94] Kolesar, P. Minimum cost replacement under markovian deterioration. *Operations Research*, 12 (9), p. 694–706, 1966.
- [95] Koole, G. Optimal repairman assignment in two symmetric maintenance models. *European Journal of Operational Research*, 82, p. 295–301, 1995.
- [96] Kumamoto, H. and E.J., Henley. *Probabilistic Risk Assessment and Management for Engineers and Scientists*. IEEE Press, 1996.
- [97] Lam, C.T. and R.H., Yeh. Comparison of sequential and continuous inspection strategies for deteriorating systems. *Advanced Applied Probability*, 26, p. 423–435, 1994.
- [98] Lam, C.T. and R.H., Yeh. Optimal maintenance policies for deteriorating systems under various maintenance strategies. *IEEE Transactions on Reliability*, 43, p. 423–430, 1994.
- [99] Lam, C.T. and R.H., Yeh. Optimal replacement policies for multistate deteriorating systems. *Naval Research Logistics*, 41, p. 303–315, 1994.
- [100] Lam, Y. An optimal maintenance model for a combination of secondhand-new or outdated-updated system. *European Journal of Operational Research*, 119, p. 274–280, 1999.

- [101] Lamb, R.G. *Availability Engineering and Management for Manufacturing Plant Performance*. Prentice Hall, 1995.
- [102] Lee, C.Y. and Z.L., Chen. Scheduling jobs and maintenance activities on parallel machines. *Naval Research Logistics*, 47, p. 145–165, 2000.
- [103] Liaw, C.-F. A branch and bound algorithm for the single machine earliness and tardiness scheduling problem. *Computers and Operations Research*, 26, p. 679 – 693, 1999.
- [104] Luss, H. Inspection policies for a system which is inoperative during inspection periods. *AIIE Transactions*, 9, p. 189–194, 1976.
- [105] Luss, H. Maintenance policies when deterioration can be observed by inspections. *Operations Research*, 24 (2), p. 359–366, 1976.
- [106] Luss, H. An inspection policy model for production facilities. *Management Science*, 29 (9), p. 1102–1109, 1983.
- [107] McCall, J.J. Maintenance policies for stochastically failing equipment: a survey. *Management Science*, 11, p. 493–524, 1965.
- [108] Menipaz, E. Optimization of stochastic maintenance policies. *European Journal of Operational Research*, 2, p. 97–107, 1978.
- [109] Monahan, G.E. A survey of partially observable markov decision processes: Theory, models and algorithms. *Management Science*, 28, p. 1–16, 1982.
- [110] Monchy, F. *Maintenance: Méthodes et Organisation*. Technique et Ingénierie, Série Génie Industrielle.
- [111] Monchy, F. *La fonction Maintenance : Formation à la Maintenance Industrielle*. Masson, 1996.
- [112] Mooney, C. Z. *Monte Carlo Simulation*. Sage Publications, Thousand Oaks, Calif., c1997.
- [113] Moubray, J. *Reliability-centered Maintenance*. Industrial Press Inc., 1997.
- [114] Munford, A.G. Comparison among certain inspection policies. *Management Science*, 27 (1), p. 260–267, 1981.
- [115] Murthy, D.N.P. and E., Asgharizadeh. Optimal decision making in maintenance service operation. *European Journal of Operational Research*, 116, p. 259–273, 1999.
- [116] Murthy, D.N.P., A., Atrens, and J.A., Eccleston. Strategic maintenance management. *Journal of Quality in Maintenance Engineering*, 8 (4), p. 287–305, 2002.
- [117] Nakagawa, T. Periodic inspection policy with preventive maintenance. *Naval Research Logistics Quarterly*, 31 (1), p. 33–40, 1984.
- [118] Nakajima, S. *Introduction to TPM*. Productivity Press, Cambridge, MA, 1988.
- [119] Nakajima, S. *La Maintenance Productive Totale (TPM)*. AFNOR, Paris, 1990.
- [120] Narayan, V. The raison d'être of maintenance. *Journal of Quality in Maintenance Engineering*, 4 (1), p. 38–50, 1998.
- [121] Niederreiter, H. and P., Jau-Shyong Shiue. *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. Springer, 1994.

- [122] Okumura, S., A.K.S., Jardine, and H., Yamashina. An inspection policy for a deteriorating single-unit system characterized by a delay-time model. *International Journal of Production Research*, 34, p. 2441–2460, 1996.
- [123] Ow, P.S. and E.T., Morton. The single machine early/tardy problem. *Management Science*, 35, p. 177 – 191, 1989.
- [124] Park, K.S. Optimal number of minimal repairs before replacement. *IEEE Transactions on Reliability*, 28 (2), p. 137 – 140, 1979.
- [125] Pierskalla, W.P. and J.A., Voelker. A survey of maintenance models: the control and surveillance of deteriorating systems. *Naval Research Logistics Quarterly*, 23, p. 353–388, 1976.
- [126] Pintelton, L.M. and L., Gelders. Maintenance management decision making. *European Journal of Operational Research*, 58, p. 301–317, 1992.
- [127] Qi, X., T., Chen, and F., Tu. Scheduling the maintenance on a single machine. *Journal of the Operation Research Society*, 50 (10), p. 1071–1078, 1999.
- [128] Rausand, M. Reliability centered maintenance. *Reliability Engineering and System Safety*, 60 (2), p. 121–132, 1998.
- [129] Richet, D., M., Gabriel, D., Mallon, and G., Blaison. *Maintenance basée sur la fiabilité*. Masson, 1996.
- [130] Righter, R. Optimal policies for scheduling for repairs and allocating heterogenous servers. *Journal of Applied Probability*, 33, p. 537–547, 1996.
- [131] Rinaldi, G. and A., Sassano. On a job scheduling problem with different ready times: Some properties and a new algorithm to determine the optimal solution. Rapporto dell'Ist. di Automatica dell'Universita di Roma e del C.S.S.C.C.A.-C.N.R.R. 77-24, 1977.
- [132] Rubinstein, R. Y. *Simulation and the Monte Carlo Method*. J. Wiley, New-York, c1981.
- [133] Sandve, K. and T., Aven. Cost optimal replacement of monotone, repairable systems. *European Journal of Operational Research*, 116 (2), p. 235–248, 1999.
- [134] Sarker, R. and A., Haque. Optimization of maintenance and spare provisioning policy using simulation. *Applied Mathematical Modelling*, 24, p. 751–760, 2000.
- [135] Scarf, P.A. On the application of mathematical models in maintenance. *European Journal of Operational Research*, 99 (3), p. 493–506, 1997.
- [136] Segawa, Y., M., Ohnishi, and T., Ibaraki. Optimal minimal-repair and replacement problem with age dependent cost structure. *Computers and Mathematics with Applications*, 24, p. 91–101, 1992.
- [137] Seshadri, S. Determination of aggregate preventive maintenance programs using production schedules. *Computers and Industrial Engineering*, 14, p. 193–200, 1988.
- [138] Seshadri, S. and M., Pinedo. Optimal allocation of resources in a job shop environment. *IIE Transactions*, 31, p. 195–206, 1999.
- [139] Shahani, A.K. and S.B., Newhold. An inspection policy for the detection of failure in the weibull case. *The Quality Engineer*, 36 (9), p. 8–10, 1972.

- [140] Sherif, Y.S. and M.L., Smith. Optimal maintenance models for systems subject to failure: A review. *Naval Research Logistics Quarterly*, 28, p. 47–74, 1981.
- [141] Sherwin, D. A review of overall models for maintenance management. *Journal of Quality in Maintenance Engineering*, 6 (3), p. 138–164, 2000.
- [142] Sielersleben, J. Dynamically optimized replacement with a markovian renewal process. *Journal of Applied Probability*, 18, p. 641–651, 1981.
- [143] Smith, A.M. *Reliability-Centred Maintenance*. McGraw-Hill, New York, NY, 1993.
- [144] Smith, D.R. Optimal repair of a series system. *Operations Research*, 26, p. 653–662, 1978.
- [145] Sridhar, S. Determination of aggregative maintenance programs using production scheduling. *Computers and Industrial Engineering*, 14, p. 193–200, 1988.
- [146] Stadje, W. and D., Zuckerman. Optimal repair policies with general degree of repair in two maintenance model. *Operations Research Letters*, 11, p. 77–80, 1992.
- [147] Suzuki, T., editor. *TPM In Process Industries*. Productivity Press, Portland, Oregon, 1994.
- [148] Tadikamalla, P.R. An inspection policy for the gamma failure distributions. *Journal of the Operational Research Society*, 30, p. 77–80, 1979.
- [149] Tahara, A. and T., Nishida. Optimal replacement policy for minimal repair model. *Journal of the Operations Research Society of Japan*, 18, p. 113–124, 1974.
- [150] Tajiri, M. and F., Gotoh. *TPM Implementation*. McGraw-Hill, New York, NY, 1992.
- [151] Thomas, L.C. A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering*, 16, p. 297–309, 1986.
- [152] Tilquin, C. and R., Cléroux. Periodic replacement with minimal repair at failure and adjustment costs. *Naval Research Logistics Quarterly*, 22, p. 243–254, 1982.
- [153] Tsang, A.H.C. Strategic dimension of maintenance management. *Journal of Quality in Maintenance Engineering*, 8 (1), p. 7–39, 2002.
- [154] Tsang, A.H.C., A.K.S., Jardine, J.D, Campbell, and J.V., Picknell. *Reliability-Centred Maintenance: A Key to Maintenance Excellence*. City University of Hong Kong, Hong Kong, 2000.
- [155] Turco, F. and P., Parolini. A nearly optimal inspection policy for productive equipment. *International Journal of Production Research*, 21, p. 515–528, 1984.
- [156] Valdez-Flores, C. and R.M., Feldman. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics Quarterly*, 36, p. 419–446, 1989.
- [157] Valente, J.M.S. and R.A.F.S., Alves. An exact approach to early/tardy scheduling with release dates. Working Paper 129, Faculdade de Economica do Porto, Portugal, 2003.
- [158] Valente, J.M.S. and R.A.F.S., Alves. Heuristics for the early/tardy scheduling problem with release dates. Working Paper 130, Faculdade de Economica do Porto, Portugal, 2003.
- [159] Van der Duyn Schouten, F.A. and C.S., Tapiero. O.R. models for maintenance management and quality control. *European Journal of Operational Research*, 82, p. 211–213, 1995.

- [160] Van der Duyn Schouten, F.A. and S.G., Vanneste. Maintenance optimization of a production system with buffer capacity. *European Journal of Operational Research*, 82, p. 323–338, 1995.
- [161] van Dijkhuizen, G. and A., van Harten. Optimal clustering of frequency-constrained maintenance jobs with shared set-ups. *European Journal of Operational Research*, E 99, p. 552–564, 1997.
- [162] van Harten, A. On the sequencing and clustering of maintenance jobs. *Internal Report, University of Twente, the Netherlands*, 1994.
- [163] Villemeur, A. *Sûreté de Fonctionnement des Systèmes Industriels*. Edition Eyrolles Paris, 1988.
- [164] Villemeur, A. *Reliability, Availability, Maintainability and Safety Assessment*, volume 1. J. Wiley, UK, 1992.
- [165] Wang, H. A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139, p. 469–489, 2002.
- [166] Wartenhorst, P. n parallel queuing systems with server breakdowns and repair. *Journal of the Operational Research Society*, 51, p. 145–155, 2000.
- [167] Weinstein, L. and C.H., Chung. Integrating maintenance and production decision in a hierarchical production planning. *Computer and Operations Research*, 26, p. 1059–1074, 1999.
- [168] Wijnmalen, D.J.D. and J.A.M., Hontelez. Review of a markov decision algorithm for optimal inspections and revisions in a maintenance system with partial information. *European Journal of Operational Research*, 62, p. 96–104, 1992.
- [169] Wilderman, R.E. and R., Dekker. Dynamic influences in multi-component maintenance. *Quality and Reliability Engineering International*, 13, p. 199–207, 1997.
- [170] Wilderman, R.E., R., Dekker, and A.C.J.M., Smit. A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, 99, p. 530–551, 1997.
- [171] Zuckerman, D. Replacement models under additive damages. *Naval Research Logistics Quarterly*, 24, p. 549–558, 1977.
- [172] Zuckerman, D. Inspection and replacement policies. *Stochastic Processes and their Applications*, 7, p. 141–151, 1978.
- [173] Zuckerman, D. Optimal policies for the case where the damage process is a one-sided lévi process. *Stochastic Processes and their Applications*, 7, p. 141–151, 1978.
- [174] Zwingelstein, G. *Diagnostic des défaillances*. Hermès, Paris, 1995.
- [175] Zwingelstein, G. *La Maintenance basée sur la Fiabilité*. Hermès, Paris, 1996.