

**République Algérienne Démocratique et Populaire**  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

UNIVERSITÉ BATNA-II  
Faculté de Technologie  
Département de Génie Industriel

---

## THÈSE

PRÉSENTÉE AU

**Laboratoire d'Automatique et de Productique (LAP)**

Pour obtenir le titre de diplôme de

**DOCTORAT EN SCIENCES DE L'UNIVERSITÉ BATNA**

Spécialité : Génie Industriel

Par

**Younes BAHMANI**

Magister en Génie Industriel- Université de Batna

**Thème**

---

*Optimisation multicritère de  
l'ordonnancement des activités de la  
production et de la maintenance intégrées  
dans un atelier Job Shop*

---

Soutenu le : 15 / 04 / 2017, devant le jury composé de :

Pr. M.Dj MOUSS	MCA. Université de Batna - 2	Président
Dr. H. SMADI	MCA. Université de Batna - 2	Rapporteur
Pr. A. DIB	MCA. Université d'Oum el Bouaghi	Examinateur
Pr. Z. SARI	MCA. Université de Tlemcen	Examinateur
Dr. F. GHEDJATI	MCA. Université de Reims, France	Examinatrice
Dr. H. KALLA	MCA. Université de Batna -2	Examinateur

---

Avec mes profonds sentiments d'amour et de reconnaissance, je dédie ma thèse à l'hommage  
de l'esprit saint et chaleureux, du cœur irremplaçable et de l'effort acharné de ma défunte  
maman, source divine et éternelle d'amour et tendresse.

À mon père, à qui j'exprime un profond amour et respect

À ma chère épouse, la rose de ma vie

À mes filles Bia-Asma et Romaissa

À mes beaux parents

À mes frères et sœurs

À ma famille

---

---

## REMERCIEMENTS

Ce présent travail a été effectué au sein de Laboratoire de recherche d'Automatique et Productique (LAP), département Génie Industriel faculté de technologie université Batna-II.

C'est un agréable devoir pour moi d'exprimer ma très vive reconnaissance à Monsieur Docteur Hacene SMADI, Directeur de l'institut d'hygiène et sécurité Batna-II pour m'avoir guidé durant toute l'élaboration de cette thèse avec le sérieux et la compétence qui les caractérisent. Qu'ils trouvent ici le témoignage de notre très profonde gratitude.

Je suis honorée par la présence de Messieurs, Monsieur Mohamed Djamel MOUSS, Professeur à l'Université de Batna et Chef de département de Génie industriel, d'avoir accepté de présider le jury.

J'adresse mes remerciements aux membres de jury, Monsieur Abderrahmane DIB, Professeur à l'Université d'Oum Labbouagui et Doyen de la faculté des sciences et des sciences appliquées, Monsieur SARI Zaki, Professeur à l'Université de Tlemcen et Directeur du Laboratoire de Productique de Tlemcen (MELT), Madame Docteur Fatima GHEDJATI Maître de conférences en Informatique, Laboratoire CReSTIC, Université de Reims Champagne-Ardenne. et Monsieur Hamoudi KALLA, Maître de conférences au département d'informatique université Batna – 2, de m'avoir fait l'honneur d'examiner ce travail et de porter leur critique objective et instructive remplie d'expérience.

Mes remerciements vont également à Mme. Leila Hayet MOUSS, Professeur, directrice du Laboratoire de recherche LAP et Présidente du comité Scientifique de Département Génie Industriel, pour tous ce qu'elle a porté pour notre formation et pour ses encouragements.

Enfin, Je tiens à remercier tous les enseignants de département Génie Industriel et particulièrement Dr. BENBRAHIM Meriem, Dr. BENZOUAI Messaoud et Mr. LATRECH Khaled, pour leurs aide, leurs encouragements et leurs soutien moral dans des moments très difficiles.

---

## Sommaire

Introduction Générale .....	1
<b>Chapitre 1. Problème d’ordonnement dans le système de production</b>	
Introduction .....	6
1. Système de production .....	7
1.1. Décomposition du système de production .....	8
1.2. Rôle et importance de la gestion de production .....	9
1.3. Niveaux hiérarchiques de la gestion de production .....	10
1.3.1. Le niveau stratégique .....	10
1.3.2. Le niveau tactique .....	10
1.3.3. Le niveau opérationnel .....	10
1.4. Rôle de l’ordonnement en gestion de production .....	11
2. Notions préliminaires en ordonnancement .....	13
2.1. Définition de l’ordonnement .....	13
2.2. Les opérations .....	14
2.3. Les ressources .....	15
2.3.1. Ressources renouvelables .....	15
2.3.2. Ressources consommables .....	15
2.4. Les contraintes .....	16
2.4.1. Contraintes temporelles .....	17
2.4.2. Contraintes liées aux ressources .....	17
2.4.3. Les critères .....	20
3. Notation des problèmes d’ordonnement .....	23
3.1. Le champ Alpha .....	23
3.2. Le champ Bêta .....	25
3.3. Le champ Gamma .....	26
4. Typologie des ateliers .....	27
4.1. Les ateliers à une ressource .....	27
4.2. Les ateliers à plusieurs ressources .....	28
4.2.1. Les ateliers de type flow-shop .....	29
4.2.2. Les ateliers de type job-shop .....	29
4.2.3. Les ateliers de type open-shop .....	30
5. Représentation des problèmes d’ordonnement .....	30
5.1. Le diagramme de Gantt .....	30
5.2. Graphe disjonctif .....	31
6. Caractéristiques générales des ordonnancements .....	32

6.1. Ordonnancement admissible.....	32
6.2. Ordonnancements semi-actifs.....	33
6.3. Ordonnancements actifs.....	34
6.4. Ordonnancements sans délais.....	35
6.5. Ordonnancement statique/dynamique.....	35
7. Problème d’ordonnancement Job Shop (JSSP).....	36
7.1. Description du problème d’ordonnancement job shop.....	36
8. Complexité des problèmes d'ordonnancement.....	38
8.1. Complexité du problème d’ordonnancement job shop.....	40
Conclusion .....	41

**Chapitre 2. Ordonnancement de la maintenance et de la production.**

Introduction.....	43
1. Évolution de la maintenance.....	44
2. Définition de la maintenance .....	45
3. Place de la maintenance dans le système de production .....	45
4. Types de maintenance.....	46
4.1. Les activités de la maintenance.....	47
4.2. La maintenance corrective.....	48
4.3. La Maintenance Préventive.....	49
4.3.1. La Maintenance Préventive systématique.....	49
4.3.2. Maintenance préventive conditionnelle.....	50
4.3.3. Maintenance préventive prévisionnelle.....	51
5. Politique de maintenance.....	52
6. Les fonctions opérationnelles d’une structure maintenace.....	54
6.1. La fonction étude de maintenance (méthodes).....	54
6.2. La fonction préparation des interventions .....	55
6.3. La fonction d’ordonnancement.....	55
6.3.1. Particularité de l’ordonnancement de la maintenance.....	56
6.4. Les objectifs de l’ordonnancement de la maintenance.....	56
6.5. La fonction réalisation des interventions.....	57
7. Coût de maintenance.....	57
7.1. Couts de maintenance corrective .....	59
7.1.1. Les coûts directs .....	59
7.1.2. Les coûts indirects.....	59
7.2. Couts de maintenance préventive.....	60
7.2.1. Coût de la maintenance préventive systématique .....	60
8. Les attributs de la sûreté de fonctionnement.....	60
8.1. Fiabilité.....	61

8.2. Disponibilité.....	61
8.3. Maintenabilité .....	61
8.4. Les temps estimateur moyens.....	62
9. Lois continues usuelles utilisées en fiabilité.....	63
9.1. Loi exponentielle.....	64
9.1.1. Domaine d'utilisation. ....	64
9.1.2. Caractéristiques de Loi exponentielle.....	65
9.2. Loi de Weibull .....	65
9.2.1. Domaine d'utilisation de la loi de Weibull. ....	65
9.2.2. Expressions mathématiques caractéristiques de la loi de Weibull, .....	66
10. Les stratégies d'ordonnancement production / maintenance.....	68
10.1. Stratégie d'ordonnancement séparé.....	68
10.2. Stratégie d'ordonnancement séquentiel.....	69
10.3. Stratégie d'ordonnancement intégré.....	70
10.3.1. Définition de l'intégration.....	70
10.3.2. Politique d'intégration.....	70
11. Intégration des fonctions production et maintenance .....	71
11.1. La situation conflictuelle des fonctions production et maintenance.....	72
11.2. Typologie de l'indisponibilité.....	73
11.3. Le processus de négociation de la production et de la maintenance.....	75
12. Intérêts d'un ordonnancement de la production et de la maintenance .....	77
Conclusion .....	77

### **Chapitre 3. Problème d'Optimisation Multicritères**

Introduction.....	79
1. Problèmes d'optimisation .....	80
2. Problèmes d'optimisation monocritère.....	81
2.1. Variables de décision .....	81
2.2. Espace décisionnel et espace objectif .....	81
2.3. Contraintes .....	81
2.4. Hiérarchisation des méthodes d'optimisation monocritère .....	82
3. Problèmes d'optimisation multicritères.....	83
3.1. Définition.....	83
3.2. Approches de résolution de point de vue décideur.....	84
3.3. Hiérarchisation des méthodes d'optimisation multicritère.....	86
4. Approches de résolution des problèmes d'optimisation multicritères.....	86
4.1. Approches non Pareto.....	87
4.1.1. Approches scalaires.....	88

4.1.2. Approches non-scalaires.....	89
4.2. Approches Pareto.....	91
4.2.1. Notion préliminaires.....	92
4.2.2. L'équilibre souhaité entre l'intensification et la diversification.....	94
4.2.3. Mécanisme de sélection Pareto (Ranking).....	94
4.2.4. L'élitisme.....	95
4.2.5. Méthodes de diversification.....	96
4.2.6. Difficultés de maintenir la diversité sur la frontière de Pareto.....	99
4.3. Difficultés des méthodes d'optimisation multicritère.....	100
4.3.1. Multi-modalité.....	101
4.3.2. Isolation d'un optimum.....	101
4.3.3. Tromperie.....	101
4.4. Choix de la stratégie de résolution.....	102
4.5. Évaluation des solutions d'optimisation multicritère.....	103
5. Les principales métaheuristiques pour l'optimisation multicritères.....	103
5.1. Les techniques Non-élitiste.....	104
5.1.1. Multiple Objective Genetic Algorithm (MOGA).....	104
5.1.2. Non dominated Sorting Genetic Algorithm (NSGA).....	105
5.1.3. Niche Pareto Genetic Algorithm (NPGA).....	107
5.2. Les techniques élitistes.....	108
5.2.1. Strength Pareto Evolutionary Algorithm (SPEA).....	108
5.2.2. Méthode Pareto Archived Evolution Strategy (PAES).....	110
5.2.3. Pareto Envelope based Selection Algorithm (PESA).....	111
5.2.4. Region Based Selection (PESA II).....	112
5.2.5. Non dominate Sorting Genetic Algorithm II (NSGAI).....	113
5.2.5.1. Calcul de la distance de crowding.....	115
5.3. Comparaison des approches multicritères.....	115
6. Problème de complexité.....	116
6.1. La complexité algorithmique.....	116
6.2. Algorithmes polynomiaux et Algorithmes exponentiels.....	117
6.3. Les différentes classes de complexité.....	117
6.3.1. La réduction de Turing.....	117
6.3.2. La classe P.....	117
6.3.3. La classe NP.....	117
Conclusion.....	118

**Chapitre 4. Optimisation multicritère d'un JSSP intégrée des activités de la maintenance et de la production**

Introduction.....	121
1. Description du problème d'ordonnancement.....	123
2. Formulation du MCJSSP.....	124
2.1. Le Makespan.....	124
2.2. Coût de production.....	125
2.2.1. Coût de matière première.....	125
2.2.2. Coût de stockage.....	125
2.2.3. Coût de détention.....	126
2.2.4. Coût de lancement.....	126
2.2.5. Coût de fabrication.....	126
2.2.6. Coût de pénalité.....	127
2.2.7. Le coût de production total.....	129
2.3. Coût de maintenance.....	129
2.3.1. Calcul de la période de disponibilité opérationnelle.....	130
2.3.2. Coût d'avance et de retard des interventions.....	132
2.3.3. Coût d'interventions de maintenance.....	134
2.3.4. Coût total de maintenance.....	134
2.4. Contraintes.....	134
2.4.1. Contraintes de positivité.....	134
2.4.2. Contrainte d'unicité d'exécution.....	135
2.4.3. Contrainte d'enchaînement.....	135
2.4.4. Les contraintes de non chevauchement.....	135
2.4.5. Contrainte d'indisponibilité.....	136
3. Application de NSGA-II au MCJSSP.....	138
3.1. Description de l'algorithme NSGA-II.....	138
3.2. Les opérateurs proposés pour NSGA-II.....	140
3.2.1. Codage des solutions.....	140
3.2.2. Génération aléatoire de la population initiale.....	141
3.2.3. Opérateurs génétiques.....	142
3.3. Expérimentations et Résultats.....	146
3.3.1. Comparaison du makespan.....	147
3.3.2. Analyse du makespan et des coûts de production.....	149
3.3.3. Analyse du makespan, coûts de production et coûts de maintenance.....	150
Conclusion.....	154
<b>Conclusion générale et perspectives.....</b>	<b>156</b>
<b>Références Bibliographiques.....</b>	<b>160</b>



## TABLE DES FIGURES

Figure (1.1) : systèmes de production. ....	8
Figure (1.2) : Organisation hiérarchique d'un système de gestion de production.....	11
Figure (1.3) : les sous-fonctions de l'ordonnancement dans l'atelier.....	12
Figure (1.4) : Notations relatives à l'exécution d'une tâche.....	14
Figure (1.5) : Typologie des problèmes d'ordonnancement par ressource.....	16
Figure (1.6) : contraintes disjonctives.....	18
Figure (1.7): Illustration de la contrainte cumulative.....	19
Figure (1.8): Relation entre les critères d'optimisation.....	23
Figure (1.9) : les types d'atelier.....	24
Figure (1.10) : Modèle machine unique.....	27
Figure (1.11) : ordonnancement d'une séquence de jobs sur machines parallèle.....	28
Figure (1.12) : ordonnancement d'une séquence de jobs en Flow Shop.....	29
Figure (1.13): ordonnancement d'une séquence de jobs en Job Shop.....	29
Figure (1.14) : classement suivant la relation machines/atelier.....	30
Figure (1.15) : Diagramme de Gantt.....	31
Figure (1.16) : Graphe disjonctif.....	31
Figure (1.17) : Diagramme de Venn.....	32
Figure (1.18) : Ordonnancement admissible.....	33
Figure (1.19) : Solution valide sans attente.....	34
Figure (2.1) : Évolution de la maintenance depuis 1940, Vrignat et al. (2012).....	45
Figure (2.2) : Politiques de maintenance et évènements révélateurs.....	47
Figure (2.3) : Graphe caractéristique de la maintenance corrective.....	48
Figure (2.4) : Graphe caractéristique de la maintenance préventive systématique.....	50
Figure (2.5) : Graphe caractéristique de la maintenance préventive conditionnelle.....	51
Figure (2.6) : Schématisation de la maintenance préventive prévisionnelle.....	51
Figure (2.7) : Choix d'une politique de Maintenance.....	52
Figure (2.8) : Évolution des coûts de défaillance.....	54
Figure (2.9) : Les trois fonctions opérationnelles de la maintenance, Monchy (2003).....	54
Figure (2.10) : Structure hiérarchique des objectifs de l'ordonnancement de la maintenance.....	57
Figure (2.11) : équilibre maintenance corrective/ préventive.....	58
Figure (2.12) : Durées caractéristiques des indicateur de Fiabilité, Maintenabilité et Disponibilité.....	63
Figure (2.13) : Passage d'un état de bon fonctionnement à l'état de panne.....	63
Figure (2.14) : Courbe en baignoire.....	64
Figure (2.15) : Représentation graphique de la loi exponentielle.....	65
Figure (2.16) : La représentation graphique de la loi Weibull.....	67

## Table des Figures

Figure (2.17) : Ordonnement Séparé.....	69
Figure (2.18) : Ordonnement Séquentiel.....	69
Figure (2.19) : Ordonnement Intégré.....	71
Figure (2. 20) : Processus de négociation production / maintenance.....	75
Figure (3. 1) : Minimums locaux et minimum globaux d'une fonction à une variable.....	82
Figure (3. 2) : Classification générale des méthodes d'optimisation monocritère.....	83
Figure (3. 3) : Problème d'optimisation multi-objectif.....	84
Figure (3. 4) : Classification générale des méthodes d'optimisation multicritère.....	87
Figure (3.5) : Approche d'agrégation.....	89
Figure (3.6) : Schéma de fonctionnement de VEGA.....	90
Figure (3.7) : Les niveaux de préférence dans la relation de dominance.....	93
Figure (3.8) : Optimalité locale au sens de Pareto.....	93
Figure (3.9) : Optimalité globale au sens de Pareto.....	93
Figure (3.10) : Front de Pareto.....	94
Figure (3.11) : Optimalité globale au sens de Pareto.....	94
Figure (3.12) : Représentation du point idéal et du point Nadir.....	94
Figure (3.13) : Illustration de la convergence et la diversité en multiobjectif.....	94
Figure (3.14) : le Ranking.....	95
Figure (3.15) : Un ensemble convexe.....	100
Figure (3.16) : Un ensemble concave.....	100
Figure (3.17) : Discontinuité due aux contraintes.....	100
Figure (3.18) : Discontinuité due à la forme du front.....	100
Figure (3.19) : Front Multimodal.....	101
Figure (3.20) : Fonction Multimodal.....	101
Figure (3.21) : Problèmes trompeurs.....	102
Figure (3.22) : Schéma de fonctionnement de NSGA.....	106
Figure (3.23) : Fonctionnement général de l'algorithme SPEA.....	109
Figure (3.24) : Schéma de fonctionnement de PESA.....	112
Figure (3.25) : Exemple de sélection par tournoi hypercube.....	113
Figure (3.26) : Schéma de fonctionnement de NSGA-II.....	114
Figure (3.27) : Calcul d'un représentant (Crowding).....	114
Figure (3.28) : Comparaison des performances des approches d'optimisation multicritère.....	115
Figure (4, 1) : pénalités d'avance.....	128
Figure (4, 2) : pénalités de retard.....	128
Figure (4.3) : dates d'intervention souhaitée et période d'intervention.....	130
Figure (4.4) : cas de $O_{i'j'} < Om_{kh}$ et $Om_{kh} < O_{ij}$ .....	137
Figure (4.5) : cas de $Om_{kh} < O_{ij}$ .....	137
Figure (4.6) : cas de $O_{ij} < Om_{kh}$ avec retard de $Om_{kh}$ .....	138

## Table des Figures

---

Figure (4.7) : cas de $O_{ij} < Om_{kh}$ avec avance de $O_{ij}$ .	138
Figure (4.8) : Construction d'ordonnement à partir d'un chromosome codé à base d'opérations.	141
Figure (4.9) : croisement combiné production maintenance.	145
Figure (4.10) : Mutation basée sur permutation des jobs.	146
Figure (4.11) : Comparaison des résultats (makespan) avec BKS.	147
Figure (4.12) : Comparaison des gains résultats RPD1 et RPD2.	149
Figure (4.13) : Comparaison des déviations RPD3 et RPD4.	150
Figure (4.14) : évolution du makespan sans intégration des opérations de maintenance.	151
Figure (4.15) : évolution des coûts de production avec l'intégration des opérations de maintenance.	151
Figure (4.16) : déviation de makespan et production avec intégration des opérations de maintenance.	151
Figure (4.17) : les solutions Pareto optimales des critères obtenues par NSGA-II.	152
Figure (4.18): Progress of the solutions found by NSGA-II.	152
Figure (4.19) : Diagramme de GANTT de l'instance Ft06 sous contrainte d'indisponibilité.	153

## TABLE DES TABLEAUX

Tableau (1.1) : Paramètres d'ordonnancement.....	26
Tableau (1.2) : Critères d'ordonnancement.....	27
Tableau (1.3) : Gamme opératoire des jobs.....	29
Tableau (4.1) : résultats des expériences.....	149

## TABLE DES ALGORITHMES

Algorithme (1.1) : Générateur d'ordonnancement Actif.....	35
Algorithme (1.2) : Générateur d'ordonnancement Sans Délais.....	35
Algorithme (3.1) : Assignation du rang de Pareto.....	95
Algorithme (3.2) : Algorithme MOGA.....	105
Algorithme (3.3) : Algorithme de la fonction de sélection.....	108
Algorithme (3.4) : Strength Pareto Evolutionary Algorithm.....	109
Algorithme (3.5) : Algorithme de PAES.....	111
Algorithme (3.6) : Algorithme de la Fonction de test (c, m, archive).....	111
Algorithme (3.7) : Pareto Envelope based Selection Algorithm.....	112
Algorithme (3.8) : Non dominate Sorting Genetic Algorithm II.....	114
Algorithme (4.1) : Pseudo-code de NSGA-II.....	139
Algorithme (4.2) : Population initiale.....	142
Algorithme (4.3) : Algorithme de sélection.....	143
Algorithme (4.4) : principe de croisement LOX.....	144
Algorithme (4.5) : principe de croisement pour maintenance par croisement.....	145

---

# **INTRODUCTION GÉNÉRALE**

# Introduction Générale

La conjoncture économique contemporaine est caractérisée par la globalisation des marchés, l'émergence de nouvelles formes de concurrence, sur lesquels les exigences et les attentes des clients deviennent de plus en plus forte en termes de qualité, de coût et de délais de mise à disposition des produits. Ces défis se renforcent par le développement rapide de nouvelles technologies des biens matériels, d'information et de la communication qui permettent une relation directe inter-entreprises et entre entreprises-clients.

De ce fait, il est crucial pour persister, être efficace et performant, l'entreprise doit passer obligatoirement par une gestion de production plus fiable. En effet, une telle gestion doit organiser le fonctionnement du système de production et de mieux gérer ses différentes opérations, et d'assurer un niveau acceptable de synergie entre ses fonctions dont les objectifs généralement antagonistes. Ainsi, l'amélioration de la production dépend étroitement de l'ordonnancement. Car, la détermination de ce qu'il faut faire, quand le faire et avec quelle ressource, peut être décisive quant à la réduction du temps et des coûts de la production. [Olivier et al. 2014].

Par conséquent, l'ordonnancement dans les ateliers consiste à prévoir l'enchaînement de l'ensemble des opérations élémentaires nécessaires à la réalisation des ordres de fabrication sur les ressources de la production tout en tenant compte des contraintes internes et externes. En effet, l'ordonnancement est une composante cruciale dans la planification et la gestion de la production, particulièrement dans l'industrie manufacturière.

La résolution d'un problème d'ordonnancement débute par l'identification et la modélisation des contraintes à respecter et les critères à optimiser, qui sont fortement liés à la nature des produits à fabriquer, des ressources requises et au type d'atelier. En second lieu, on doit chercher une méthode de résolution adéquate, celle-ci est en grande partie définie en fonction de la taille et de la complexité du problème d'ordonnancement.

En outre, être performant, passe obligatoirement par le maintien de l'outil de production en bon état de fonctionnement, qui est toujours une préoccupation majeure dans un monde industriel où les notions de Just-à-temps, de réactivité, de coûts et de qualité ont une grande importance. De ce fait, il est vital de pouvoir s'appuyer sur un système de production disponible et performant à tout instant. Ces multiples contraintes, de gestion, techniques, commerciales et financières, ont des incidences majeures sur les fonctions qui ont des liens très étroits avec la production au niveau opérationnel. Ces

liens intrinsèques de la production avec ces fonctions limitrophes et particulièrement la maintenance, qui a la charge d'assurer la disponibilité des moyens de production par des interventions correctives et/ou préventives, font que leurs objectifs respectifs peuvent être en conflit. En effet, la maintenance et la production sont deux fonctions qui agissent sur les mêmes ressources. Cependant, leurs activités respectives sont ordonnancées d'une manière indépendante, sans tenir compte de cette contrainte.

Les machines jusqu'à une époque non lointaine, sont considérées continuellement disponibles ou éventuellement durant un horizon de temps. Dès lors, la planification de la maintenance n'est jamais prise en considération lors de l'établissement de l'ordonnancement de la production. Toutefois, l'arrêt des machines pour maintenance est planifié séparément des tâches de production et ne s'insère pas dans le programme de production, sauf en cas de panne.

Cette hypothèse de disponibilité des ressources est contestée par la réalité des entreprises [Ma et al. 2010 et Kamran 2013]. En effet, Une machine peut être hors service soit pour une intervention planifiée (maintenance préventive), où l'une des opérations de production ou de maintenance est maintenue à sa date prévue et l'autre sera retardée ou avancée. Soit suite à l'occurrence d'une défaillance (maintenance corrective) l'opération de production est retardée après l'accomplissement de l'opération de maintenance et la remise en service de la machine.

Comme l'indisponibilité est néfaste quant à la productivité, sa maîtrise et prise en considération lors de l'ordonnancement de la production, demeurent l'un des facteurs décisifs dans un système de fabrication, visant le maintien de ses machines en bon état de fonctionnement. De ce fait les coûts de production avec prise en considération des tâches de la maintenance dans l'ordonnancement et de la production restent largement inférieurs et le coût engendré par une panne peut être largement supérieur à celui d'un arrêt prévu de la production.

Un problème d'ordonnancement intégrant les opérations de la production et les opérations de la maintenance est soumis à plusieurs facteurs de complexité, à savoir les facteurs liés à la typologie des systèmes de production (type d'atelier), ceux liés aux types de tâche de maintenance et à l'horizon de planification, aux nombres et aux compétences des ressources humaines de maintenance.

Dans cette thèse, on s'intéresse particulièrement à un problème d'ordonnancement intégrant les tâches de la maintenance et de la production, avec comme objectif le respect des contraintes intrinsèques au problème. Les données du plan de production sont fournies par la gestion de production, tandis que le plan de maintenance est caractérisé par un ensemble de tâches de maintenance préventive systématique, où chaque opération de maintenance est liée à une machine et nécessite la présence d'un intervenant pour maintenance, l'ensemble de ces opérations forme un travail



(job) qui sera ajouté aux jobs de production. Alors, l'objectif fixé est de minimiser simultanément le temps de séjour maximum des produits dans l'atelier (Makespan), les coûts de la production, et de la maintenance.

Nous sommes donc, face à un problème d'optimisation multicritères d'un ordonnancement qui intègre les activités de la production et de la maintenance dans un atelier job shop, sous contraintes de sûreté de fonctionnement des équipements, pour assurer la disponibilité et la pérennité de l'outil de production. Or, l'indisponibilité des machines est due aux interventions pour maintenance préventive systématique (MPS). Les périodes sont calculées, en exploitant le modèle probabiliste de Weibull. La date de début de l'intervention peut être glisser dans une fenêtre de temps et le dépassement de cette fenêtre est soumis à des pénalités d'avance ou de retard. De même, la production est sous la contrainte de juste à temps, dans le sens où elle est pénalisée en fonction de l'avance et de retard par rapport aux dates d'échéance des produits.

La complexité du problème est dû au type d'atelier, au nombre et à la nature des opérations de la production et de la maintenance et au nombre de critères à optimiser. Dans ce cas une approche métaheuristiques (algorithme NSGA-II) peut assurer la résolution même partiellement, de ce type de problème.

Cette thèse est organisée en quatre chapitres comme suit :

Afin de positionner le sujet dans son cadre général. Nous décrivons le contexte général de l'ordonnancement et ses problèmes.

Dans le premier chapitre, nous présentons les notions de base concernant la production et la gestion de production, en mettant l'accent sur le rôle de l'ordonnancement au sein de ces systèmes. La suite du chapitre est réservée au cadrage de la problématique générale de l'ordonnancement par la présentation des points essentiels : définitions, typologies, modélisations, critères d'optimisation et organisation des ateliers. Enfin, La classification des problèmes d'ordonnancement nous amène à étudier la complexité de ces derniers en général et celle de l'atelier Job Shop en particulier.

Le deuxième chapitre est dédié à la nécessité de la mise en œuvre d'une relation de coopération entre les services de production et de la maintenance. Nous commençons par la description de la fonction maintenance : sa typologie, ses concepts ainsi que sa structure. Ensuite, nous présentons les spécificités de l'ordonnancement de la maintenance. Nous décrivons, les différents coûts de maintenance et les attributs de la sûreté de fonctionnement. Enfin, les situations conflictuelles entre la fonction production et la fonction maintenance, ainsi que, les intérêts que porte la politique d'intégration des activités de ces deux fonctions sont mis en exergue.

Le troisième chapitre est consacré à l'introduction des problèmes d'optimisation, pour présenter en outre la définition, la formulation et les approches de résolution des problèmes d'optimisation multicritères. Nous mettons l'accent sur les notions liées à l'optimisation multicritères. Un intérêt particulier sera porté sur l'étude des algorithmes multicritères basés sur la dominance de Pareto. Dans ce cas il ne s'agit pas de décrire en profondeur chacun de ces modèles mais de faire une synthèse de recensement des méthodes les plus répondues en précisant leurs principales caractéristiques.

Dans le dernier chapitre, nous commençons par la description du problème à étudier, suivie d'une formulation mathématique du problème d'ordonnancement job shop multicritères avec contrainte d'indisponibilité. Puis, nous étudions en détail les différents coûts de production et de maintenance nécessaire à la détermination des critères à optimiser. Ensuite, nous adaptions l'algorithme NSGA-II à notre problème et les résultats expérimentaux sont interprétés.

Enfin, cette thèse se termine par une conclusion ainsi que des perspectives de recherche envisagées.

---

## **CHAPITRE 1.**

# **ORDONNANCEMENT DE LA PRODUCTION DANS UN SYSTÈME DE PRODUCTION**

# Chapitre 1.

## Problème d'ordonnancement dans le système de production

---

***Résumé :** Les problèmes d'ordonnancement constituent un domaine très vaste de la recherche opérationnelle et de la gestion de la production. Ils sont présents dans tous les secteurs d'activité de l'économie, depuis l'industrie manufacturière jusqu'à l'informatique. C'est pour cette raison qu'ils sont d'un intérêt théorique et pratique toujours renouvelé. Par l'ordonnancement les entreprises visent à améliorer leur efficacité en termes de coût de production et de délai de livraison.*

*Résoudre un problème d'ordonnancement consiste à planifier, dans le temps l'exécution des tâches en leur attribuant les ressources matérielles et/ou humaines nécessaires de manière à satisfaire un ou plusieurs critères préalablement définis, tout en respectant un certain nombre de contraintes. Les études de la complexité ont montrés que la majorité des instances de problème d'ordonnancement job shop (JSSP) sont NP-difficiles. Dans ce chapitre, nous commençons par une introduction à l'ordonnancement en général et à l'atelier de JSSP en particulier, puis, les définitions et les différentes typologies et notations utiles pour le reste de nos travaux. Une classification des problèmes d'ordonnancement ainsi qu'un rappel sur les notations de complexité sont abordés. Enfin, une description du JSSP constitue le centre de nos intérêts a été abordé.*

---

### Introduction

Dans le domaine de l'industrie où la compétitivité est féroce, les entreprises doivent faire face à l'augmentation de la concurrence, à la pression de plus en plus forte de leur environnement (clients et concurrents) pour un renouvellement rapide des produits, mais aussi une augmentation de l'offre vis à vis de la demande. L'accroissement de la productivité en réduisant les coûts est, aujourd'hui, un objectif majeur dans toutes les entreprises. Les systèmes de production sont caractérisés par leur dynamique et leur imprévision, les tâches souvent de caractère complexe sont soumises à des contraintes de temps et d'exigence. Alors, le système de production se donne de nouveaux objectifs à atteindre. Ceux-ci peuvent être obtenus par une meilleure utilisation des ressources. En particulier, la productivité peut être fortement affectée par l'ordonnancement des opérations sur les ressources (machines).

L'ordonnancement est une allocation, dans le temps, des ressources disponibles aux différentes tâches, dans le but d'optimiser un ou plusieurs objectifs. Il occupe une place particulière et joue un rôle privilégié dans la gestion informatisée des flux de production au sein de l'entreprise, s'inscrivant dans des niveaux de décision à la fois tactique et opérationnel, [Giard 1991], [Esquirol et al. 1999]. C'est généralement le point de rencontre entre un système hiérarchisé et informatisé de production et le système de production lui-même. La gestion de production crée les ordres de fabrication qui déterminent globalement ce que doit être fait dans un horizon de temps généralement imposé.

L'importance et la richesse de la problématique de l'ordonnancement est due aux différentes interprétations que peuvent prendre les ressources et les tâches. Ainsi, les ressources peuvent être des machines dans un atelier, des pistes de décollage et d'atterrissage dans un aéroport, des équipes dans un terrain de construction, des processeurs dans les ordinateurs, etc. Alors que les tâches peuvent être des opérations dans un processus de production, le décollage et l'atterrissage dans un aéroport, les étapes d'un projet de construction, l'exécution d'un programme informatique, etc. Les différentes tâches sont caractérisées par un degré de priorité et un temps d'exécution. Les ressources quant à elles, sont caractérisées par une capacité, des temps de réglage, etc. Plusieurs objectifs peuvent être visés selon le problème d'ordonnancement à traiter : minimiser le temps total d'exécution des tâches, le nombre des tâches devant être exécutées après leur date d'expiration et les délais d'attente, etc.

Dans un premier temps, il est nécessaire de replacer l'ordonnancement dans le cadre général des systèmes de production. Nous rappelons, des notions de base concernant la production et la gestion de la production, en mettant l'accent sur le rôle de l'ordonnancement au sein de ces systèmes. Nous présentons le problème de l'ordonnancement en précisant sa définition et les différents éléments qui le déterminent. Nous présentons par la suite les techniques de modélisation des problèmes d'ordonnancement. Une description des différentes organisations d'ateliers qui déterminent des problèmes d'ordonnancement spécifiques a été présentée. Enfin le formalisme d'ordonnancement dans un atelier job shop est évoqué.

### 1. Système de production

Un système de production est composé d'un ensemble de ressources organisées qui interagissent et interfèrent dans le but de produire des biens ou services. Comme tout système de la vie réelle, les systèmes de production sont finis. Ils ne peuvent jamais écouler une charge illimitée de travail. Ce qui les limite, ce sont les ressources présentes en quantité et/ou capacité limitées. On qualifie donc ces systèmes de « système de production à partage de ressources ».

La gestion de la production est l'ensemble des activités qui participent à la conception, la planification des ressources (matérielles, financières, ou humaines), l'ordonnancement, l'enregistrement des activités de production et le contrôle des activités de la production de l'entreprise. Son objectif est d'optimiser les processus de valeur ajoutée en améliorant de manière continue les flux allant des fournisseurs aux clients. L'ensemble de ces activités doit être réalisé dans le respect des procédures établies (implicitement ou explicitement) par l'entreprise et tenir compte à la fois de la qualité de ses produits ou services, mais aussi de la sécurité de ses salariés ou de son environnement. Pour mener à bien ces différentes tâches, les entreprises s'aident d'outils informatiques, de la gestion de la production assistée par ordinateur aux progiciels de gestion intégrés (PGI ou ERP) en passant par la supervision. Le rôle de la gestion de la production a évolué avec le temps du fait des changements qu'a connu l'environnement économique.

La production peut être définie comme étant un processus de transformation, issu d'un système de production, qui transforme des matières premières et/ou des composants dits brut au sens général, en produit fini plus élaborés et de valeur économique plus élevée.

Parmi les ressources on distingue essentiellement les équipements (machines, outils, moyens de transport, moyens informatiques, ...), les moyens humains qui permettent le bon déroulement du processus de transformation, produits à différentes étapes de fabrication (matières premières,

produits semi-finis, produits finis, ...), les entrepôts de matières ou des aires de stockage et les finances.

Les équipements de production sont les machines de production, permettant d'effectuer des opérations de transformation, les machines de manutention, permettant de transporter des pièces dans l'atelier (robots, chariots mobiles, tapis roulant, transstockeurs, ...), et les machines de contrôle de qualité. Les deux dernières peuvent être considérées comme des machines de production spéciales ou fictives.

### 1.1. Décomposition du système de production

Les systèmes de production industrielle se sont considérablement diversifiés et compliqués, [Blondel 2004], [Vacher 2000]. En effet, ils peuvent se décomposer en trois sous-systèmes : le sous-système physique de production, le sous-système d'information et le sous-système de décision Figure (1.1), qui s'intègrent en vue d'assurer la pérennité et la compétitivité de l'entreprise.

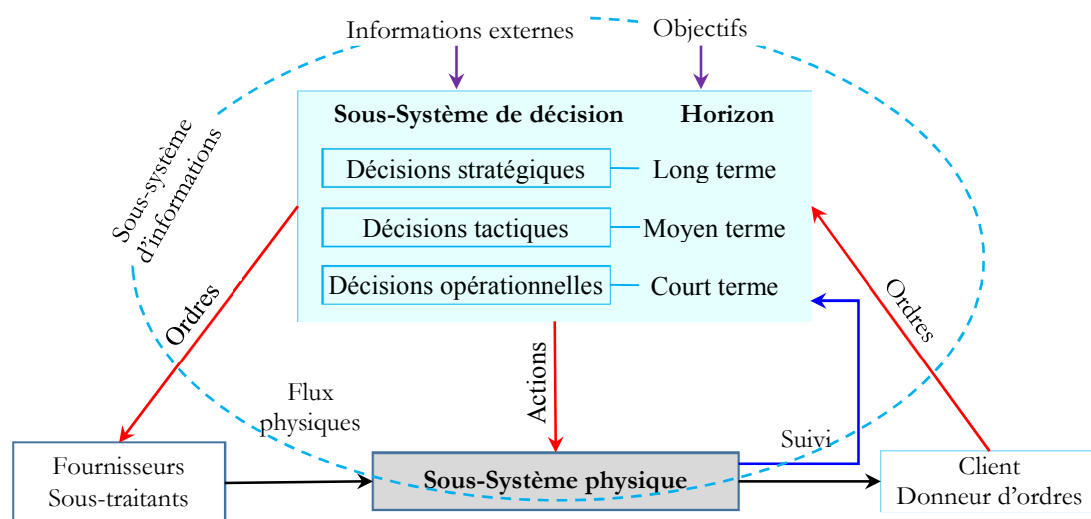


Figure (1.1) : systèmes de production

- Le sous-système physique de production englobe toutes les différentes ressources nécessaires pour la transformation des matières premières en produits finis.
- Le sous-système de décision contrôle le système physique de production à travers l'organisation des différentes activités en prenant des décisions basées sur les données transmises par le système informationnel.
- Le sous-système d'information joue le rôle d'interface entre le système de décision et le système de production d'une part. D'autre part, il intervient à l'intérieur du sous-système de décision, pour la gestion des informations utilisées lors de la prise de décision. Son rôle est de collecter, stocker et transmettre des informations de différents types, [Fontanili 1999].

- Les sous-systèmes décisionnel et informationnel sont couramment appelés système de gestion de production. Ils traitent les fonctions rattachées directement à la production à savoir, les achats, les approvisionnements, la planification, la gestion des ressources, la maintenance, etc. [Doumeingts 1984], [Roboam 1988]. La commande du système de gestion au sous-système physique est caractérisée par la transformation des informations à caractère commercial en ordres de fabrication et ordres d'approvisionnements. Le système de gestion de production reçoit un feedback sous forme d'information de suivis du sous-système physique, qui sert effectivement à piloter ce dernier.

### 1.2. Rôle et importance de la gestion de production

La gestion de la production est la mise en application de méthodes et techniques dans le but d'accomplir la transformation des matières en produits fini. Elle se résume en la combinaison de ressources, parmi lesquelles les moyens matériels (les machines), les moyens humains (le personnel par qualification) et les matières (matières premières, matières consommables) dans un planning avec pour but d'assurer la fabrication du produit en qualité et en quantité définies.

Dans un environnement économique devenu aussi concurrentiel que le nôtre, les enjeux financiers sont cruciaux. Le prix de vente des produits dépend de plus en plus de la demande du marché et reste très influencé par la concurrence. Afin de rester compétitive et surtout garantir une marge bénéficiaire convenable sur la vente de leurs produits, les entreprises industrielles ont pour principal recours la réduction du coût de production. Le champ d'action de la gestion de la production dans l'entreprise est vaste, couvre de nombreuses activités et interpelle les professionnels de différents domaines de formation.

Les contraintes rencontrées sont de divers ordres :

- Financières (produire à un coût optimal), coût des matières et consommables, coût de stockage des encours et des produits semi ouvrés, coût de gestion des magasins, coût des heures de travail supplémentaires, coût des arrêts faisant partie intégrante du coût de revient, maîtriser ces derniers est aussi une garantie pour la commercialisation des produits finis,
- Temporelles (produire dans les délais, assurer une livraison juste à temps), éviter les ruptures de stocks, éviter le gonflage des stocks de produits finis. Car cela a une incidence directe sur la satisfaction de la clientèle (pertes de commandes) ou sur le coût de revient du produit finis dû aux coûts supplémentaires du stockage,
- Mécaniques (maintenance préventives et gestion des temps d'arrêt), anticiper sur les pannes et prévoir des solutions alternatives en cas d'arrêt d'une machine,



- Qualité (produire avec le moins de défaut possible, le moins de déchets), un produit de bonne qualité participe à la fidélisation de la clientèle, véhicule l'image de marque de l'entreprise.
- Planification : assurer une circulation continue des flux, détecter et supprimer les goulets d'étranglement dans le circuit de production. Il s'agit aussi à ce niveau de définir un plan de production, de définir les gammes opératoires, d'ordonnancer les opérations, et enfin de gérer la répartition des tâches durant tout le processus de fabrication.

### 1.3. Niveaux hiérarchiques de la gestion de production

La gestion de la production s'effectue par un ensemble de décisions qui peuvent être hiérarchisées suivant un ordre temporel, dont plusieurs activités peuvent être attribuées au système de gestion. La gestion de production est un ensemble de processus qui permet de mener à bien la fabrication de produits à partir d'un ensemble de données et de prévisions, [Vacher 2000]. En fait, la gestion de production s'occupe d'un ensemble de problèmes liés à la production tels que la gestion des données, la planification, le contrôle (suivi) de la production, la gestion des stocks, la prévision, l'ordonnancement etc.

En général, trois niveaux hiérarchiques de la gestion de production sont couramment retenus, à savoir les niveaux ; stratégique, tactique et opérationnel, [Giard 1988], [Vacher 2000], [Fontanili 1999], [Letouzey 2001].

#### 1.3.1. Le niveau stratégique

Il s'agit de la formulation de la politique et la stratégie de l'entreprise à long terme, qui porte essentiellement sur la gestion des ressources durables, afin que celles-ci soient en mesure d'assurer la pérennité de l'entreprise sur plusieurs années.

#### 1.3.2. Le niveau tactique

Il s'agit de décisions à moyen terme qui s'effectuent sur plusieurs mois. Elles assurent la liaison entre le niveau stratégique et le niveau opérationnel. L'objectif est de produire au moindre coût pour satisfaire la demande prévisible, en s'inscrivant dans le cadre fixé par le plan stratégique de l'entreprise.

#### 1.3.3. Le niveau opérationnel

C'est le niveau d'exécution du système qui sert à une gestion quotidienne pour faire face à la demande du jour le jour, dans le respect des décisions tactiques. Il génère des décisions à court et à très court terme qui déterminent l'ordonnancement et le pilotage.

La Figure (1.2) présente l'interaction qui existe entre les activités du système de gestion et les niveaux de planification de ces activités.

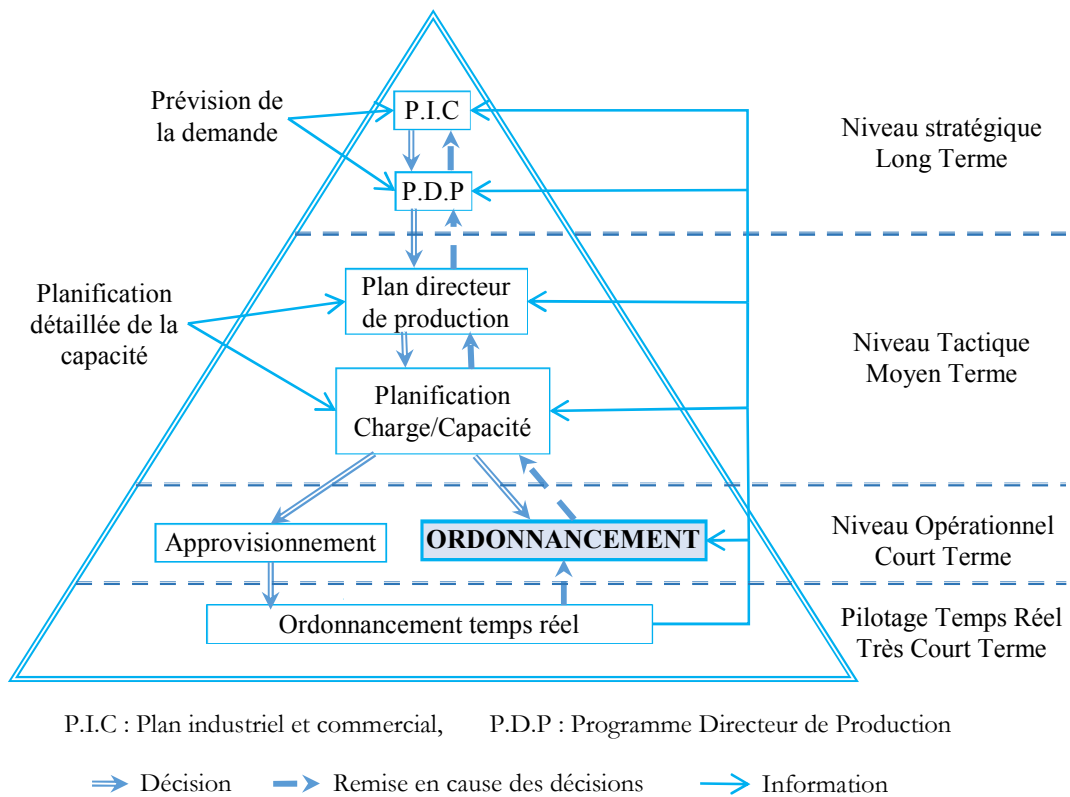


Figure (1.2) : Organisation hiérarchique d'un système de gestion de production

Nos travaux sont positionnés au niveau opérationnel de l'organisation hiérarchique du système de production. En général, à ce niveau l'ordonnancement généré est statique et prévisionnel. Or, la prise en considération des aléas au niveau pilotage temps réel, rend l'ordonnancement dynamique.

En réalité, la gestion de production doit répondre aux fameuses questions : que produire ? Quand ? Comment ? Par qui ? et À quel coût ?

Les questions quand ? par qui ? et À quel coût ? constituent l'axe de nos travaux.

### 1.4. Rôle de l'ordonnancement en gestion de production

L'ordonnancement d'atelier se fait dans un contexte particulier. Ce contexte est défini par la gestion de la production utilisée par l'entreprise. Il occupe une place particulière et joue un rôle privilégié dans la gestion informatisée des flux de production au sein de l'entreprise, s'inscrivant dans des niveaux de décision à la fois tactique et opérationnel, [Lopez et al. 1999], [Jeffrey 2006]. C'est généralement le point de rencontre entre un système hiérarchisé et informatisé de production et le système de production lui-même.

En pratique, l'occurrence des événements aléatoires endogènes : pannes de machines, grèves, ... et exogènes : commandes imprévues et prioritaires, ... est fréquente et fortement probable. En effet, la mise à jour de l'ordonnancement devient nécessaire, par conséquent les problèmes d'ordonnancement sont traités aux niveaux inférieurs.

L'ordonnancement traduit l'ensemble des décisions de fabrication définies par le programme directeur de production en instructions d'exécution détaillées destinées au lancement au contrôle et au pilotage à court terme de l'activité des postes de travail. À l'issue de la fonction ordonnancement, on obtient un calendrier qui assure une affectation optimale des tâches sur les ressources disponibles, en précisant la durée et la date d'exécution de chacune d'elles, tout en respectant certaines contraintes et en optimisant un ou plusieurs fonctions objectifs.

La fonction ordonnancement se décompose en trois sous-fonctions ; la première sous fonction s'occupe de l'élaboration des ordres de fabrication (OF) c'est-à-dire transformation des informations du programme directeur de production (suggestion de fabrication) en ordres de fabrication. Dans la seconde, la préparation du programme d'atelier : ceci dit, la détermination en fonction des ordres de fabrication et de la disponibilité des ressources le calendrier prévisionnel de fabrication. Enfin dans la troisième, le lancement et le suivi des opérations de fabrication sont appliqués, Figure (1.3).

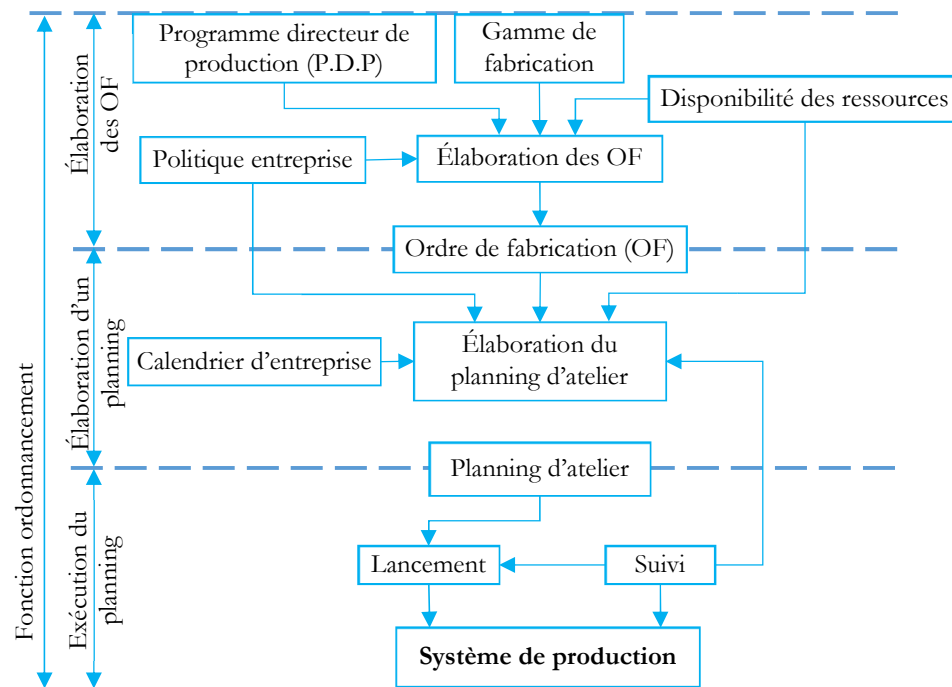


Figure (1.3) : les sous-fonctions de l'ordonnancement dans l'atelier, [Jeffrey 2004]

## 2. Notions préliminaires en ordonnancement

Les problèmes d'ordonnancement trouvent leur application dans divers champs d'application, notamment dans le secteur industriel, informatique, économique, etc., raison pour laquelle, ont été largement étudiés ces dernières décennies. Pour une description plus détaillée de la problématique de l'ordonnancement, le lecteur peut se référer aux ouvrages : [Lopez et al., 2001], [Pinedo 2012], [Kołodziej 2012], [Vanhoucke 2013], [Mauergauz 2013], [Emmons et al. 2013], [Framinan et al. 2014], [Cai et al. 2014], [Pinedo 2016].

### 2.1. Définition de l'ordonnancement

Plusieurs définitions de l'ordonnancement sont proposées dans la littérature, nous citons :

- 1) Ordonner c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution. [Carlier et al. 1988]
- 2) L'ordonnancement consiste à organiser dans le temps la réalisation d'un ensemble de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement, etc.) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises par les tâches. Un ordonnancement décrit l'ordre d'exécution des tâches et l'allocation des ressources au cours du temps, afin de satisfaire un ou plusieurs critères d'optimisation. [Esquirol et al. 1999], [Harrath 2003]

Le mot ordonnancement désigne soit une solution au problème d'ordonnancement, soit par abus de langage, le processus ayant conduit à la détermination de cette solution. Les tâches (ou opérations), les ressources, les contraintes et les critères constituent ses éléments principaux.

Essentiellement, Il s'agit de la prise de type de décision : La première est l'affectation de chacune des opérations à ordonner à une ressource adéquate, parmi un parc de ressources disponible. La deuxième est la détermination du séquençement des opérations attribuées à une même ressource en précisant leur ordre de passage ainsi que leurs dates de début d'exécution.

L'allocation des opérations aux ressources est réalisée en tenant compte des contraintes temporelles définies par la planification, des charges et de la disponibilité des ressources imposées par la programmation prévisionnelle de la production, des contraintes de succession des opérations imposées par la gamme de fabrication, et vise souvent à satisfaire un ou plusieurs objectifs (fonction coût), exprimés au sein d'un ou plusieurs critères de performance.

Les paragraphes suivants précisent ces notions de tâche, ressource, objectif, et introduisent quelques notations.

## 2.2. Les opérations

Une tâche, une activité ou une opération notée  $O_{ij}$  d'un job  $J_i$  de  $n_i$  opérations  $J_i = \{O_{ij}\}_{j=1}^{n_i}$  est une entité élémentaire de travail localisée dans le temps par une date de début  $s_i$  et de fin  $c_i$ . L'exécution de l'opération nécessite une durée positive  $p_{ij} \geq 0$ .

Une opération est dite préemptive si sa réalisation peut être interrompue  $s_{ij} + p_{ij} \leq c_{ij}$ . Si non, elle est dite non-préemptive,  $s_{ij} + p_{ij} = c_{ij}$

Certaines notations relatives à l'exécution des tâches sont définies ainsi :

- les dates de début au plus tôt (earliest starting time) ou realise date : c'est la date avant laquelle l'opération ne peut pas commencée.  $r_{ij} = est_{ij} = \min(s_{ij})$ .
- les dates de début au plus tard (latest starting time) : c'est la date après laquelle l'opération ne peut pas commencée.  $lst_{ij} = \max(s_{ij})$ .
- les dates de fin au plus tôt (earliest completion time) : c'est la date avant laquelle l'opération ne peut pas terminée  $ect_{ij} = \min(c_{ij})$ .
- les dates de fin au plus tard (latest completion time) ou due date : c'est la date avant laquelle l'opération doit terminée  $d_{ij} = lct_{ij} = \max(c_{ij})$ .
- La fenêtre de temps d'une tâche (time window) : désigne l'intervalle durant lequel la tâche est potentiellement exécutée  $[est_{ij}, lct_{ij}]$  ou  $[r_{ij}, d_{ij}]$ .
- La partie obligatoire (compulsory part) : c'est l'intervalle durant lequel la tâche est obligatoirement exécutée  $[lst_{ij}, ect_{ij}]$

Pour une tâche non préemptive, ou non morcelable, Une variable unique  $s_i$  ou  $c_i$  suffit pour définir une tâche non préemptive de durée fixe Figure (1.4).

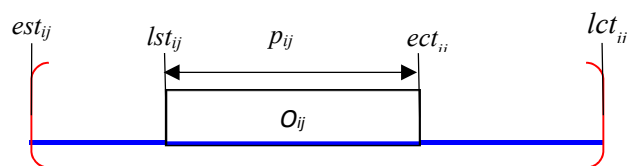


Figure (1.4) : Notations relatives à l'exécution d'une tâche.

Une particularité importante des problèmes considérés est que la durée des tâches n'est pas fixée mais bornée :  $p_{ij} \in [p_{ij}, p]$ . Cela exprime notamment des incertitudes sur la connaissance exacte de la durée d'exécution des tâches qui ne sera réellement connue qu'à l'exécution effective, cette durée est ainsi qualifiée de contingente. Ce caractère incertain de la durée justifie l'utilisation de deux variables temporelles  $s_{ij}$  et  $c_{ij}$  pour représenter l'exécution de l'opération  $O_{ij}$ .

### 2.3. Les ressources

L'exécution des différentes tâches requièrent un ensemble de moyens technique, humain et financier. Ces moyens sont des ressources indispensables à la réalisation des tâches durant des intervalles de disponibilité. Elles sont en réalité disponibles en quantité limitée et sujettes à des pannes et nécessite des interventions de maintenance. La disponibilité est généralement exprimée par une capacité propre à chaque ressource  $k$  notée  $Q_k$ . On distingue deux types de ressources : les ressources renouvelables et les ressources consommables [Gargouri et al. 1999].

#### 2.3.1. Ressources renouvelables

Les ressources sont dites renouvelables, si après son utilisation la ressource est à nouveau disponible avec la même capacité initiale, c'est-à-dire elle est toujours disponible en même quantité (les hommes, les machines, l'espace, l'équipement en général, etc.). On distingue par ailleurs, principalement parmi ces ressources, les ressources disjonctives (ou non partageables) qui ne peuvent exécuter qu'une tâche à la fois (machines-outils, robot manipulateur, etc.) et les ressources cumulatives (ou partageables) qui peuvent être utilisées par plusieurs tâches en même temps (équipe d'ouvriers, poste de travail, etc.). Les problèmes d'ordonnancement à ressources disjonctives couvrent une classe importante d'applications qu'on appelle les problèmes d'atelier ou de machines.

#### 2.3.2. Ressources consommables

Les ressources sont dites consommable, si après son utilisation la ressource est disponible avec une capacité inférieurs ou nulle, (matières premières, budget, etc.)

Suivant la nature des ressources, une typologie des problèmes d'ordonnancement peut être présenté sur la Figure (1.5) [Mouhoub 2011].

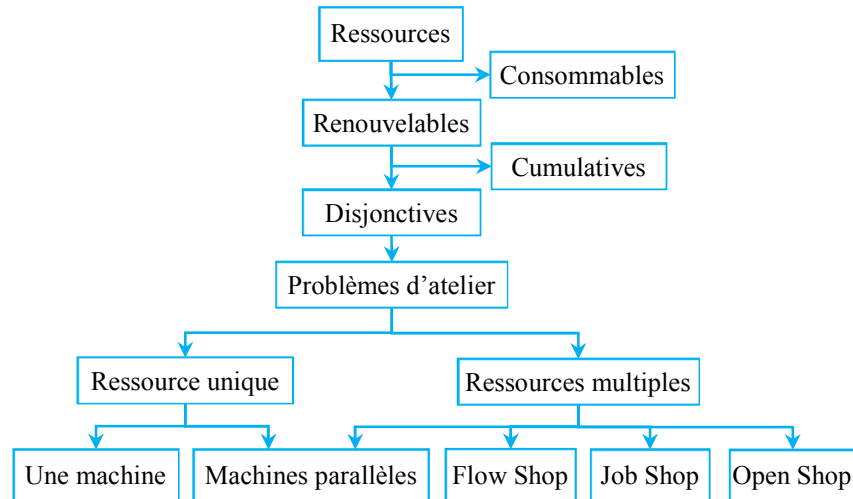


Figure (1.5) : Typologie des problèmes d'ordonnancement par ressource

Dans la pratique, la flexibilité de certaines ressources peut conduire à la prise en compte des temps de reconfiguration supplémentaires, qui peuvent dépendre de la tâche à exécuter, mais aussi sur l'état de la ressource elle-même, qui est souvent liée à des tâches précédentes exécutées sur cette ressource. Cela signifie que les temps de réglage doivent être pris en considération, en fonction de la séquence choisie. L'examen de ces temps de préparation est vital dans la pratique, car ils peuvent avoir un sérieux impact sur l'ordonnancement. Il encourage le regroupement des produits identiques en lots sur chaque machine afin de minimiser le temps total de reconfiguration de la machine. En outre, la durée des tâches ne sont pas toujours connus, mais peut dépendre de la quantité de moyens affectés à leur exécution qui à son tour peut être liée à la vitesse ou les performances de la ressource.

### 2.4. Les contraintes

La notion de contrainte est attachée à un ensemble de variables de décision. Elle exprime les restrictions sur les domaines de ces variables de décision. Il existe deux classifications des contraintes. La première comporte les contraintes temporelles. Ces dernières sont de deux type : relative aux décisions de localisation des tâches dans le temps (variables d'ordonnancement si et/ou ci), ou bien, relatives aux décisions d'affectation des tâches sur les ressources (variables d'affectation) [Saulé 2008]. La deuxième distingue les contraintes endogènes qui sont liées directement au système de production et à ses performances, des contraintes exogènes indépendantes du système, [Kacem 2003]. Les contraintes permettent de rendre l'ordonnancement plus réaliste.

### 2.4.1. Contraintes temporelles

Les contraintes temporelles ou contraintes de précédences, permettent d'exprimer les interdépendances temporelles entre tâches, (exigences technologiques, logistiques, etc.). Ces contraintes sont classiquement formalisées sous la forme d'une inégalité de potentiels.

#### 2.4.1.1. Les contraintes de dates limites

Les contraintes de dates limites sont imposées individuellement à chaque opération  $i$  du job. La tâche  $i$  ne peut commencer avant sa date de lancement ou de disponibilité  $r_{ij}$  (avant laquelle la tâche ne peut pas commencer) et doit être terminée avant une date d'échéance  $d_{ij}$  (avant laquelle la tâche doit être achevée).

On obtient ainsi les contraintes de potentiels suivantes :

$$s_{ij} \geq r_{ij} \quad \forall O_{ij} \in J_i \quad (1,1)$$

$$c_{ij} \leq d_{ij} \quad \forall O_{ij} \in J_i \quad (1,2)$$

En général, on réduit directement les fenêtres de temps lorsque les dates limites sont connues à l'avance [Malapert 2011].

#### 2.4.1.2. Les contraintes d'enchaînement

Une contrainte qui relie la date de début ou la date de fin de deux opérations par une relation linéaire est dite contrainte d'enchaînement ou de succession [Baptiste et al 1998]. Ces contraintes sont généralement imposées par la cohérence technologique (les gammes opératoires dans le cas d'ateliers) qui impose l'ordre d'exécution des opérations d'un même job.

La contrainte de précédence entre les deux opérations  $O_{ij}$  et  $O_{ik}$ , notée symboliquement  $O_{ij} < O_{ik}$  ( $O_{ij}$  précède  $O_{ik}$ ) est représentable par une unique inégalité de potentiel :

$$s_{ik} - c_{ij} \leq t_{ij} \quad \text{avec } t_{ij} \geq 0 \quad (1,3)$$

- $t_{ij} > 0$ , s'il s'agit d'un cas particulier où la contrainte de précédence est avec temps d'attente (de préparation, de montage, d'entretien, de refroidissement, de séchage, etc.),
- $t_{ij} = 0$ , s'il s'agit des contraintes de précédence sans temps d'attente.

### 2.4.2. Contraintes liées aux ressources

Ces contraintes traduisent la nature de disponibilité en quantité limitée des ressources. On distingue deux types de ces contraintes : liés à la nature disjonctive ou cumulative.



### 2.4.2.1. Les contraintes disjonctives

Une contrainte de disjonction, ou non-chevauchement, entre deux tâches est satisfaite si les tâches s'exécutent dans des fenêtres de temps disjointes, avec temps d'attente équation (1.4) ou sans temps d'attente équation (1.5). Cette contrainte peut par exemple exprimer l'existence de deux gammes opératoires différentes. Arbitrer une disjonction consiste à déterminer l'ordre relatif entre les tâches.

$$O_{ij} \simeq O_{uv} \Leftrightarrow (O_{ij} \leq O_{uv}) \vee (O_{uv} \leq O_{ij}) \quad (1,4)$$

$$O_{ij} \sim O_{uv} \Leftrightarrow (O_{ij} < O_{uv}) \vee (O_{uv} < O_{ij}) \quad (1,5)$$

$O_{ij}$  et  $O_{uv} \in \Phi_k$  sont à exécuter sur la même ressource.

Une précedence  $O_{ij} < O_{uv}$  est dite interdite lorsque les fenêtres de temps des tâches sont incompatibles avec la décision, ce qui rend l'autre précedence obligatoire. Une précedence  $O_{ij} < O_{uv}$  est dite obligatoire dès qu'elle est directement impliquée par les fenêtres de temps.

Arbitrer une contrainte disjonctive consiste à décider si  $O_{ij}$  sera exécutée avant après l'opération  $O_{uv}$ , équation (1.6), Figure (1.6).

$$s_{uv} \leq c_{ij} \text{ ou } s_{ij} \leq c_{uv} \quad (1,6)$$

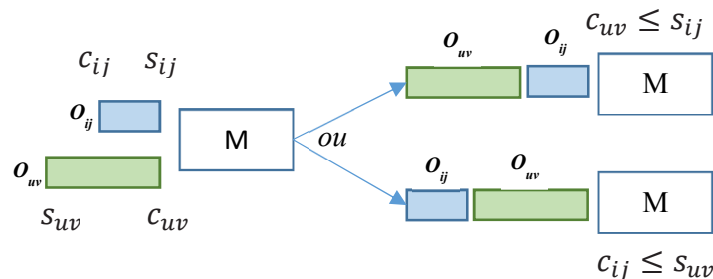


Figure (1.6) : contraintes disjonctives

### 2.4.2.2. Les contraintes cumulatives

La contrainte cumulative permet de résoudre des problèmes d'ordonnancement relatif aux plusieurs disciplines tels que : l'informatique (affectation des tâches aux processeurs), la construction (gestion et suivi d'un projet, [Creemers et al 1995]), l'industrie de production (problèmes de gestion d'ateliers [Bellone 1992], [Bisdorf 1995]), l'administration (gestion des emplois du temps, [Boizumault 1994], rotation et affectation du personnel). La contrainte cumulative exprime le fait qu'à tout instant, le total des ressources utilisées ne dépasse pas une certaine limite fixée ou

variable  $L_{lim}$ , [Aggoun 1993]. On parle de contraintes cumulatives lorsque les tâches demandent une partie d'une ou de plusieurs ressources présentes en quantité limitée Figure (1.7).

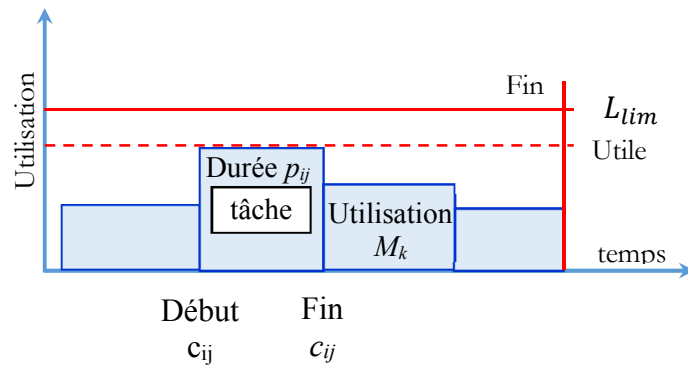


Figure (1.7): Illustration de la contrainte cumulative

$$\text{Contrainte cumulative } (s_{ij}, p_{ij}, M_k, L_{lim}, c_{ij})$$

Débuts est la liste des dates de début  $s_{ij}$  des tâches, date de fin  $c_{ij}$  Durées la liste de leurs durées  $p_{ij}$ , Ressources la liste des quantités de ressource  $M_k$  qu'elles utilisent, Limite le maximum de ressources disponibles, et Fin la date de fin de toutes les tâches.

### 2.4.2.3. Contraintes endogènes et exogènes

#### 2.4.2.3.1. Les contraintes endogènes

Elles constituent des contraintes directement liées au système de production et à ses performances tel que :

- Capacité des machines et des moyens de transport,
- Les dates de disponibilité des machines et des moyens de transport,
- Les séquences des opérations à exécuter (les gammes opératoires).

#### 2.4.2.3.2. Les contraintes exogènes

Elles sont indépendantes du système de production, mais imposées par son l'environnement :

- Les retards possibles accordés pour certains clients,
- les dates de fin de fabrication du produit au plus tard ; imposées généralement par les commandes,
- les priorités de quelques commandes et de quelques clients,

### 2.4.3. Les critères

Il est crucial lorsqu'on aborde un problème d'ordonnancement de définir un certain nombre d'objectifs à atteindre. Il s'agit donc d'optimiser une ou plusieurs fonctions d'évaluation en respectant un certain nombre de contraintes. L'optimisation dans les ateliers s'accroît principalement sur le triptyque coût, qualité et délais.

Les critères constituent les objectifs qu'on souhaite optimiser. En fonction du nombre d'objectifs on parle d'optimisation monocritère (on s'intéresse à l'optimisation d'un seul critère) et d'optimisation multi-objectifs (on s'intéresse à l'optimisation de plusieurs critères). L'optimisation s'exprime par une fonction de minimisation ou maximisation de ces critères. On trouve dans la littérature plusieurs types de critères, classés en deux catégories. Dans la première les critères sont soit de performance régulière ou de performance non-régulière [Kacem 2003]. Dans la seconde les critères sont de trois types : critères liés aux dates de fin de livraison, critères liés aux volumes des encours et en fin, critères liés à l'utilisation des ressources [Hentous 1999].

#### 2.4.3.1. Les critères réguliers

Les critères réguliers constituent des fonctions décroissantes des dates d'achèvement des opérations.

- Minimisation des dates d'achèvement,
- Minimisation du maximum des dates d'achèvement,
- Minimisation de la moyenne des dates d'achèvement,
- Minimisation des retards sur les dates d'achèvement,
- Minimisation du maximum des retards sur les dates d'achèvement,
- Minimisation de la moyenne des retards sur les dates d'achèvement,
- Minimisation du temps du cycle (cas d'ordonnancement cyclique).

#### 2.4.3.2. Les critères irréguliers

Ces critères ne constituent pas des fonctions monotones des dates de fin d'exécution des opérations. Parmi ces critères :

- la minimisation des encours ;
- la minimisation du coût du stockage des matières premières ;
- flexibilité, tolérance aux pannes, vitesse de ré-ordonnancement,
- délai de fabrication, avances, retard,

- charge des machines,
- nombre de machines actives,
- temps global de stockage (attente devant les machines),
- taille des stocks, dimensionnement des lots,
- nombre de changement d'outils. [Gargouri 2003].

La satisfaction de tous les critères à la fois est souvent délicate car on se trouve, généralement, devant des situations contradictoires [Roy et al. 1993] et à la recherche de solutions à des problèmes complexes d'optimisation. Par définition, deux critères sont équivalents si une solution optimale pour l'un est aussi optimale pour l'autre et inversement. [Carlier et al. 1988].

### 2.4.3.3. Les critères liés aux dates de fin de livraison

Les critères liés aux dates de fin de livraison constituent la catégorie des critères les plus étudiés en optimisation : on retrouve le Makespan  $C_{max} = \max(c_i)$  qui représente la fin d'achèvement de toutes les tâches du problème. Les notations ci-dessus sont également utilisées pour la définition de différents problèmes d'ordonnancement liés aux dates de fin :

- $C_{\Sigma} = \sum(c_i)$  : représente la somme des dates de fin,
- $\sum w_i c_i$  : pour représenter la somme pondérée des dates de fin,
- $\bar{C}$  : pour représenter la moyenne arithmétique des dates de fin des tâches,
- $F_{max} = \max(F_i)$  : pour représenter le temps maximal de séjour des tâches,
- $L_{max} = \max(l_i)$  : Pour représenter le maximum des retards algébriques de l'ensemble des tâches,
- $T_{max} = \max(T_i)$  : Pour représenter le maximum des retards absolus de toutes les Tâches,
- $E_{max} = \max(E_i)$  : Pour représenter le maximum des temps d'avancement de toutes les Tâches,

On retrouve aussi d'autres critères comme les temps totaux, tels que  $F_{\Sigma}, L_{\Sigma}, T_{\Sigma}, E_{\Sigma}$  pour représenter respectivement pour l'ensemble des tâches, le temps total de séjour, le retard total, le retard total absolu, et l'avancement total.

### 2.4.3.4. Les critères liés aux volumes des encours

Dans cette catégorie on s'intéresse au nombre de tâches en cours d'exécution à chaque instant  $t$  et on cherche généralement à minimiser le nombre de tâches en attente ou à maximiser le nombre de tâches terminées sur la dernière machine. Pour chaque opération on définit  $t_0$  comme sa date de début d'exécution et  $p_0$  son processing time. Ainsi il est possible de définir :

- le nombre de tâches en exécution à un instant  $t$  est donné par :

$$N_p(t) = \sum_{1 \leq i \leq n} e_i(t) \quad \text{avec} \quad e_i(t) = \begin{cases} 1 & \text{si } (t_0 \leq t \leq t_0 + p_0) \\ 0 & \text{sinon} \end{cases} \quad (1,7)$$

- le nombre de tâches en attente d'exécution  $N_w$  est donné par :

$$N_w(t) = \sum_{1 \leq i \leq n} e_i(t) \quad \text{avec} \quad e_i(t) = \begin{cases} 1 & \text{si } (t_0 + p_0 \leq t \leq t_0 + 1) \\ 0 & \text{sinon} \end{cases} \quad (1,8)$$

- le nombre de tâches terminées sur la dernière machine  $N_f$  est donné par :

$$N_f(t) = \sum_{1 \leq i \leq n} e_i(t) \quad \text{avec} \quad e_i(t) = \begin{cases} 1 & \text{si } (C_i \leq t) \\ 0 & \text{sinon} \end{cases} \quad (1,9)$$

#### 2.4.3.5. Les critères liés à l'utilisation des ressources

Les critères liés à l'utilisation des ressources permettent de satisfaire un objectif donné et d'autre part d'évaluer des critères de performance. Par exemple, pour les SFP on retrouve une panoplie de critères tels que :

- **maximiser l'utilisation de** : la machine la moins gourmande en énergie, de la machine ayant un bon rendement ou de la machine la plus simple à entretenir,
- **minimiser la charge** d'une machine ou le temps d'inactivité de l'ensemble des machines,
- **minimiser l'écart moyen** d'utilisation de l'ensemble des machines, critère qui permet d'équilibrer l'utilisation des machines...

On trouve également des critères de performance tels que Hentous (1999) :

- **Putilisation moyenne** des ressources  $\bar{U}$  donnée par :

$$\bar{U} = \frac{(\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq m} p_{ij})}{(\sum_{1 \leq j \leq m} M_j \times C_{max})} \quad (1,10)$$

Ce critère permet de mettre en évidence les périodes creuses et les périodes pleines de l'utilisation du système de production et ainsi de renseigner sur les taux d'utilisation des ressources.

- **le temps d'inactivité de l'atelier**

$$I = \sum_{1 \leq j \leq m} \left( C_{max} - \sum_{1 \leq i \leq n} p_{ij} \right) \quad (1,11)$$

Correspondant à l'ensemble des tranches de temps perdus (en attente de travail (idle time)) pendant l'exécution des tâches. C'est un indicateur de l'utilisation des capacités de l'atelier qu'il faut minimiser pour obtenir une meilleure utilisation des ressources Hentous (1999).

La combinaison entre les différents critères est possible : maximisation des profits et minimisation de tous les coûts inhérents à la production. On note aussi que des similitudes et des équivalences entre critères et que des réductions permettent de déduire certains critères des d'autres. La Figure (1.8) donne les relations entre certains critères : le sens des flèches donne le sens de réduction avec la condition associée sur chaque arc.

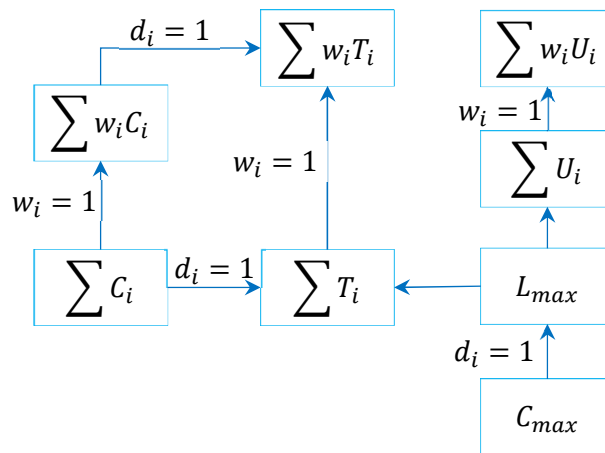


Figure (1.8): Relation entre les critères d'optimisation [Pinson 1995]

### 3. Notation des problèmes d'ordonnancement

Plusieurs notations ont été introduites dans la littérature pour spécifier un problème d'ordonnancement. Un schéma de classification proposé par Graham et al. (1979), puis repris par Blazewicz et al. (1996). Il est rapidement imposée comme faisant référence. Il permet en effet de caractériser un problème d'ordonnancement de manière précise. Cette notation structure les données d'un problème d'ordonnancement sur la base d'une notation à trois champs distincts :  $\alpha|\beta|\gamma$ . Le champ  $\alpha$  décrit l'environnement machine,  $\beta$  précise les contraintes liées aux opérations, et le champ  $\gamma$  décrit le ou les critères à optimiser.

#### 3.1. Le champ Alpha

Le champ alpha se définit comme suit :  $\alpha = \{\alpha_1, \alpha_2, \alpha_3\}$

$$\alpha_1 = \{1, P, Q, R, O, F, J\}$$

- $\alpha_1 = 1$  : Problème mono-machine. En ordonnancement de la production, certaines lignes de production peuvent se modéliser par un processeur unique. Le but est alors de rechercher l'ordre de passage des lots sur ce processeur.
  - $\alpha_1 = P$  : Un ensemble de machines parallèles identiques est susceptible de réaliser les tâches. La durée d'exécution d'un job donné est la même sur toute les machines.
  - $\alpha_1 = Q$  : Les machines disponibles pour la réalisation des jobs sont dites parallèles uniformes. Chaque machine possède une vitesse appropriée notée  $b_i$ . La durée d'exécution d'un job  $j$  sur la machine  $M_i$  notée :  $P_{ij} = \frac{p_j}{b_i}$
  - $\alpha_1 = R$  : Les machines disponibles sont parallèles quelconques. Le temps de réalisation d'une activité dépend de la machine sur laquelle sera effectuée. C'est le cas d'*Unrelated Machine*,
  - $\alpha_1 \in \{O, F, J\}$  : L'atelier est composé d'un ensemble de machines nécessaire pour la réalisation des jobs. Le type d'atelier dépend de la nature des gammes de production. L'atelier est symbolisé par O, F ou J qui signifie respectivement open shop, flow shop ou job shop.
  - $\alpha_1 = FH$  : L'atelier dédiés, cheminement unique flexible (*hybrid flow shop*). Il s'agit d'un atelier à cheminement unique, mais il peut y avoir plusieurs exemplaires de chaque machine. L'ensemble des machines d'un type est appelé « étage ». un FH est caractérisé par le nombre d'étages et le nombre de machines par étage. Le plus souvent, les durées opératoires sur un étage sont indépendantes de la machine.
- $\alpha_1 \in \mathbb{Z} +$  : Le nombre de machines  $m$  est constant,
- $\alpha_1 = o$  : Le nombre de machines  $m$  est variable,

Dans la réalité, les ateliers sont souvent des intermédiaires. La Figure (1.9) présente l'ensemble des types d'atelier. Notons que cette typologie ne prend pas en considération la recirculation (le passage répété sur la même machine) qui est néanmoins fréquente en production.

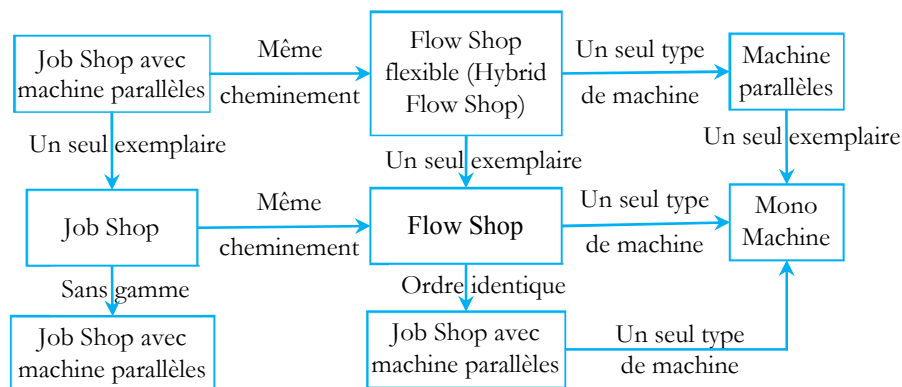


Figure (1.9) : les types d'atelier

### 3.2. Le champ Bêta

Le champ Bêta se définit comme suit :  $\beta = \{\beta_1, \beta_2, \beta_3, \beta_4, \beta_5\}$ . Il caractérise les conditions d'exécution des jobs ainsi que les états des ressources présentes dans l'atelier. Il indique l'existence ou non des contraintes de précédence entre les tâches, la possibilité de tolérer la préemption etc.

—  $\beta_1 \in \{pmtn, o, split\}$

- $\beta_1 = pmtn$  : La préemption entre les opérations est permise. Une tâche peut interrompre l'exécution d'une autre tâche, pour quelle reprendra plus tard,
- $\beta_1 = o$  : La préemption n'est pas autorisée.
- $\beta_1 = split$  : Si une tâche est décomposable en plusieurs sous tâche, alors il est autorisé de l'exécuter simultanément sur plusieurs machines.

—  $\beta_2 \in \{Prec, Tree, o\}$  ;

- $\beta_2 = Prec$  : Les tâches des jobs sont reliées par des relations de précédence,
- $\beta_2 = Tree$  : Les relations de précédence entre les tâches sont sous la forme d'un arbre,
- $\beta_2 = o$  : Pas de relations de précédence entre les tâches.

—  $\beta_3 \in \{r_j, o\}$  ;

- $\beta_3 = r_j$  : Chaque tâche  $j$  possède une date début au plus tôt «ready time » à partir de laquelle elle peut s'exécuter,
- $\beta_3 = o$  : Toutes les tâches sont exécutables à partir de la date  $s=0$ .

—  $\beta_4 \in \{p_j = 1, p_{ij} = 1, o\}$  :

- $\beta_4 = p_j = 1$  : si  $\alpha_1 = \{o, P, Q\}$  alors toutes les tâches possèdent une durée d'exécution égale à une unité de temps.
- $\beta_4 = p_{ij} = 1$  : si  $\alpha_1 = \{O, F, J\}$  alors toutes les tâches possèdent une durée d'exécution égale à une unité de temps.
- $\beta_4 = o$  : Les durées d'exécution des tâches sont toutes positives.

—  $\beta_5 \in \{S_{nsd}, R_{nsd}, R_{sd}, b_{j,(j+1)}\}$  :

- $\beta_5 = S_{nsd}$  : les machines nécessitent un temps de montage qui est indépendant des séquences,
- $\beta_5 = R_{nsd}$  : les machines nécessitent un temps de démontage qui est indépendant des séquences,
- $\beta_5 = R_{sd}$  : les machines nécessitent un temps de démontage qui dépendant des séquences,
- $\beta_5 = b_{j,(j+1)}$  : possibilité de stockage limité entre deux machines successives  $M_j$  et  $M_{j+1}$ .



### 3.3. Le champ Gamma

Le champ Gamma noté  $\gamma$  est réservé pour la description des critères. Le champ  $\gamma$  décrit le critère d'optimisation adoptés pour l'ordonnancement. Il peut prendre de nombreuses valeurs, comme il peut être une combinaison de plusieurs critères. Afin de les décrire nous introduisant les notations sur le Tableau (1.1) :

Paramètres	Désignations
$s_j$	Date de début de la tâche j (starting time)
$c_j$	Date d'achèvement de la tâche j (complétion time)
$r_j$	Date de disponibilité de la tâche j ou encore date au plus tôt (ready time).
$d_j$	Date &chue ou date au plus tard (due date) de la tâche j.
$w_j$	Poids (définit la propriété de l'exécution de la tâche j pour les clients.
$F_j = c_j - r_j$	Durée de flot indiquant la durée d'attente et d'exécution de la tâche j dans le système (flow time)
$L_j = c_j - d_j$	Avance ou retard algébrique (lateness) de la tâche j.
$T_j = \max(c_j - d_j, 0)$	Retard absolu (tardiness) de la tâche j.
$E_j = \max(d_j - c_j, 0)$	Avance absolu (earliness) de la tâche j.
$U_j = \begin{cases} 1 & \text{si } c_j > d_j \\ 0 & \text{si } \text{non} \end{cases}$	Indique si la tâche j est en retard par rapport à sa date de fin d'exécution.

Tableau (1.1) : Paramètres d'ordonnancement

Le Tableau (1.2) on résume les principaux critères qui caractérise l'ordonnancement.

Critères	Définition
$C_{max} = \max(c_j)$	Date d'achèvement de l'ordonnancement (Makespan)
$F = \sum F_j$	Flot total
$\bar{F} = \frac{1}{n} \sum F_j$	Flot moyen
$\bar{F}_w = \sum w_j F_j / \sum w_j$	Flot moyen pondéré
$L = \sum L_j$	Somme des retards algébriques
$L_{max} = \max(L_j)$	Retard algébrique maximum
$T = \sum T_j$	Somme des retards

$T_{max} = \max\{T_j\}$	Retard maximum par rapport aux dates de début au plus tard
$\bar{T} = \frac{1}{n} \sum T_j$	Retard moyen
$\bar{T}_w = \frac{\sum w_j T_j}{\sum w_j}$	Retard moyen pondéré
$E = \sum E_j$	Somme des avances
$U = \sum U_j$	Nombre de tâche en retard

Tableau (1.2) : Critères d'ordonnancement

#### 4. Typologie des ateliers

Les différents problèmes ordonnancement dépendent principalement du nombre de machines nécessaire pour réaliser chaque tâche, ainsi que sur l'enchaînement de ces dernières sur les machines. En effet, les problèmes d'ordonnancement, peuvent être classés en deux catégories. La première regroupe les problèmes pour lesquels chaque tâche nécessite une seule machine, pour la deuxième, chaque tâche demande plusieurs machines pour son exécution.

##### 4.1. Les ateliers à une ressource

Dans certains ateliers, on ne dispose que d'une ressource pour traiter un ensemble de tâches. Celle-ci ne peut traiter qu'une seule tâche à la fois. Ce type de problème consiste à déterminer la séquence optimale de passage des  $n$  tâches sur la ressource, afin d'optimiser un ou plusieurs critères donnés. À titre d'exemple, on peut citer les problèmes de minimisation du retard maximum, du nombre de tâches en retard, ou de la somme des retards. L'une des situations intéressantes où on peut rencontrer ce genre de configurations est le cas où on est devant un système de production comprenant une machine goulot qui influence l'ensemble du processus. L'étude peut alors être restreinte à l'étude de cette machine. La Figure (1.10) illustre le modèle machine unique. Malgré leur apparente simplicité, les problèmes à une machine sont NP-difficiles au sens fort [Garey et al. 1979], [Lenstra et al. 1977].

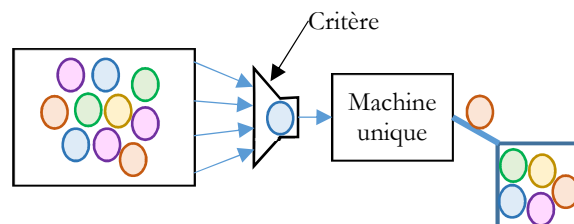


Figure (1.10) : Modèle machine unique

Les exigences du marché, et le développement des systèmes de production ont fait l'apparition des ateliers composés de machines organisées en parallèle ou non dédiées. Ce type d'atelier se caractérise par le fait que plusieurs machines sont disponibles pour l'exécution d'un travail qui n'en nécessite qu'une machine Figure (1.11). Il s'agit donc, d'une généralisation du problème à une seule machine. Il est possible de distinguer trois types d'ateliers selon la vitesse d'exécution des machines :

- 1) Ateliers à machines identiques : toute tâche peut s'exécuter sur n'importe quelle machine avec une même durée opératoire (à condition que la machine soit libre),
- 2) Ateliers à machine uniformes : chaque machine possède sa propre vitesse et ceci indépendamment de la tâche à exécuter,
- 3) Ateliers à machine indépendantes (ou non reliées) : la vitesse des machines dépend de la tâche à effectuer.

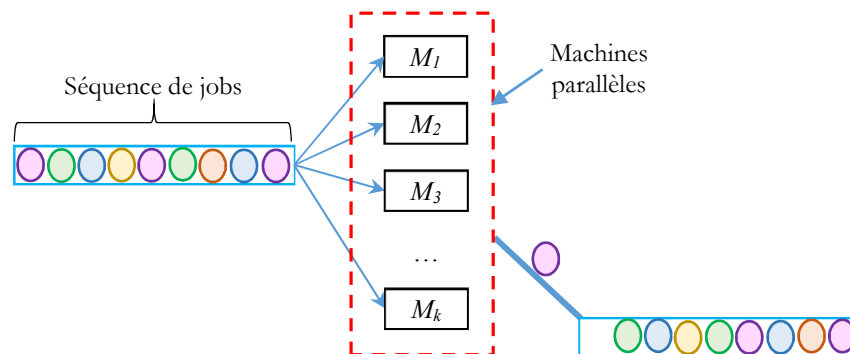


Figure (1.11) : ordonnancement d'une séquence de jobs sur machines parallèle.

### 4.2. Les ateliers à plusieurs ressources

Ces ateliers sont composés d'un ensemble de  $m$  machines dont chacune ne peut réaliser, à elle seule, l'ensemble des opérations. Suivant le mode de passage sur les différentes machines, trois types d'ateliers sont distingués. Les problèmes d'ordonnancement sont généralement classés en trois principaux modèles dépendamment du nombre d'opérations que requièrent les jobs : des modèles à une opération (machine unique et machines parallèles) et des modèles à plusieurs opérations (flow-shop, open shop et job shop).

#### 4.2.1. Les ateliers de type flow-shop

Appelés également ateliers à cheminement unique, ce sont des ateliers où une ligne de fabrication est constituée de plusieurs machines en série ; toutes les opérations de toutes les tâches passent par les machines dans le même ordre. Dans les ateliers de type flow-shop hybride, une machine peut exister en plusieurs exemplaires identiques fonctionnant en parallèle. Figure (1.12).

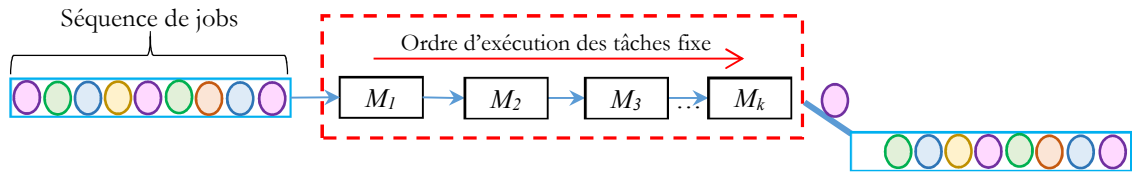


Figure (1.12) : ordonnancement d'une séquence de jobs en Flow Shop

#### 4.2.2. Les ateliers de type job-shop

Appelés également ateliers à cheminement multiple, ce sont des ateliers où les opérations sont réalisées selon un ordre défini par la gamme de fabrication prédéterminé, variant selon le job à exécuter, Tableau (1.3) et Figure (1.13). Les premiers résultats théoriques du job shop, concernant la résolution d'un problème à deux machines et plus tard à trois machines, reviennent aux années cinquante. La première définition formelle de cet atelier a paru dans l'ouvrage de Muth et Thompson 1963. Certains problèmes job shop, proposés à cette époque, ont été résolus dans les années soixante-dix. En revanche, il a fallu attendre les années quatre-vingt pour d'autres. Jusqu'à ce jour cet atelier reste le contre d'intérêt d'un nombre important de chercheurs.

Le job-shop flexible est une extension du modèle job-shop classique, sa particularité réside dans le fait que plusieurs machines sont potentiellement capables de réaliser un sous-ensemble d'opérations.

	1	2	3	4	1	2	3	4	1	2	3	4
J <sub>1</sub>	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>		10	8	4		O <sub>11</sub>	O <sub>21</sub>	O <sub>31</sub>	
J <sub>2</sub>	M <sub>2</sub>	M <sub>1</sub>	M <sub>4</sub>	M <sub>3</sub>	8	3	5	6	O <sub>22</sub>	O <sub>12</sub>	O <sub>42</sub>	O <sub>32</sub>
J <sub>3</sub>	M <sub>1</sub>	M <sub>2</sub>	M <sub>4</sub>		4	7	3		O <sub>13</sub>	O <sub>23</sub>	O <sub>43</sub>	
	Machines				Durée opératoire ( $p_{ij}$ )				Opérations			

Tableau (1.3) : Gamme opératoire des jobs

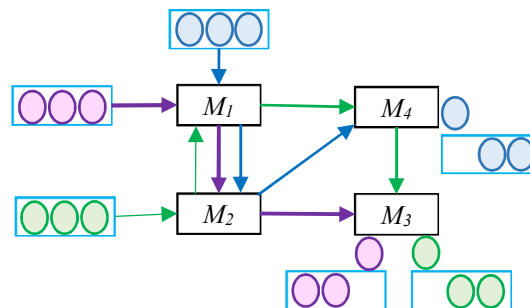


Figure (1.13) : ordonnancement d'une séquence de jobs en Job Shop

### 4.2.3. Les ateliers de type open-shop

Ce type d'atelier est moins contraint que celui de type flow-shop ou de type job-shop. Ainsi, l'ordre des opérations n'est pas fixé a priori ; le problème d'ordonnancement consiste, d'une part, à déterminer le cheminement de chaque produit et, d'autre part, à ordonnancer les produits en tenant compte des gammes trouvées, ces deux problèmes pouvant être résolus simultanément. Comparé aux autres modèles d'ateliers, l'open-shop n'est pas couramment utilisé dans les entreprises.

En général dans de tel système les espaces tampons ou Buffers entre les machines sont supposés de capacité illimitée et un job après sa fin d'exécution sur une machine doit attendre avant que son exécution ne commence sur une autre machine. Si les espaces tampons sont de capacité nulle, les jobs ne peuvent pas attendre entre deux machines consécutives et la propriété de " sans attente " est admise. La Figure (1.14) ci-dessous schématise les différents types d'ordonnancement suivant la relation machine atelier.

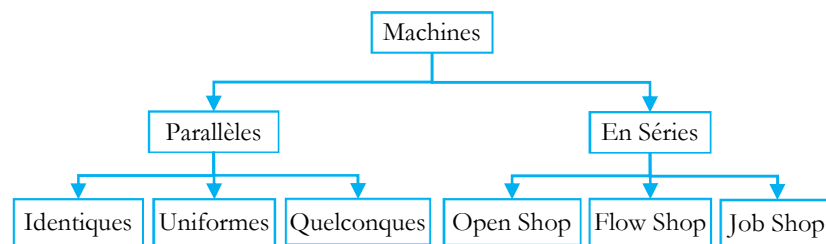


Figure (1.14) : classement suivant la relation machines/atelier

## 5. Représentation des problèmes d'ordonnancement

Toutes les solutions d'un problème d'ordonnancement d'un atelier peuvent être représentées par l'intermédiaire de deux diagrammes : diagramme de Gantt et le graphe disjonctif.

### 5.1. Le diagramme de Gantt

Le diagramme de Gantt est un outil permettant de représenter les séquences de traitement des jobs sur chaque machine requise, en précisant la date de début et de fin de chacune d'elles. Il s'agit d'un outil élaboré en 1917 par Henry L. Gantt. En effet, le diagramme de Gantt représente en ordonnée la liste des tâches, notées  $O_i$  à exécuter par les machines notées  $M_i$  et en abscisse l'échelle du temps par lequel on précise la date de début, la durée de la tâche et la date de fin de chaque tâche. À titre d'illustration une solution réalisable (avec critère Makespan) du problème Tableau (1.3) est donnée par la Figure (1.15).

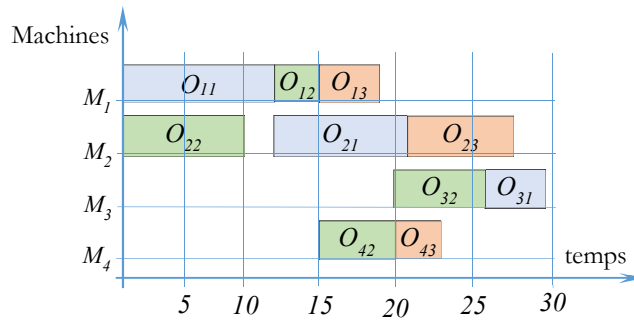


Figure (1.15) : Diagramme de Gantt

## 5.2. Graphe disjonctif

Le graphe disjonctif permet de visualiser les solutions, en plus il est considéré comme un outil de modélisation abondamment utilisé pour résoudre les problèmes d'atelier multi-machines, [Jain et al. 1999].

Le graphe disjonctif, fut proposé par [Roy et al. 1964] pour le problème du job shop. Représenté par le graphe  $G = (E, A)$  où  $A = A_1 \cup A_2$ . L'ensemble  $E$ , désignant les sommets, constitué de toutes les opérations des jobs à ordonnancer, d'une opération source (opération fictive qui marque le début d'ordonnancement) et d'une opération puits (opération fictive qui marque la fin d'ordonnancement). L'ensemble  $A_1$  désigne les arcs conjonctifs reliant chaque paire d'opérations consécutives d'un même job. L'ensemble  $A_2$  désigne les arcs disjonctifs, reliant deux opérations de jobs distincts qui utilisent une même machine. Chaque arc disjonctif consiste en une paire d'arcs d'orientations opposées tel que chaque chemin du graphe peut contenir au plus l'un d'entre eux.

Une orientation de chaque arc disjonctif est choisie de sorte à avoir un graphe acyclique, c'est à dire Sans circuit, permet d'obtenir une solution réalisable du problème d'ordonnancement. Le graphe disjonctif est dit alors arbitré (arbitrated) [Carlier et al. 1988]. À titre illustratif la Figure (1.16) montre le graphe disjonctif de l'exemple Figure (1.13). Notons que la valeur du Makespan est donnée par la longueur du plus long chemin allant de la source au puits.

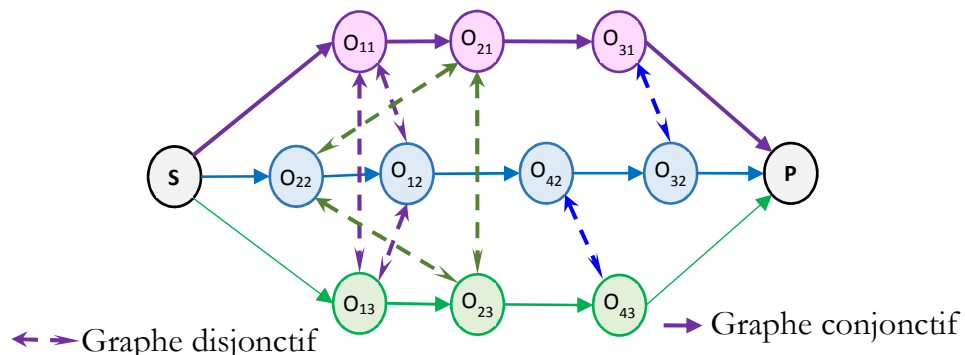


Figure (1.16) : Graphe disjonctif

## 6. Caractéristiques générales des ordonnancements

L'un des paramètres primordiaux de la génération d'un ordonnancement optimale est l'exploitation maximale du temps, car un simple décalage des opérations peut éviter les trous de temps ou temps morts. En effet, plus un ordonnancement est compact, meilleure est sa qualité [Duvivier, 2000]. Cette notion compacité est à la base de l'Il existe de différents types d'ordonnancement : admissible, actif, semi-actif et sans délais Figure (1.17) [Pinedo, 1995], [T'Kindt et al. 2006], [Esquirol at al. 1999].

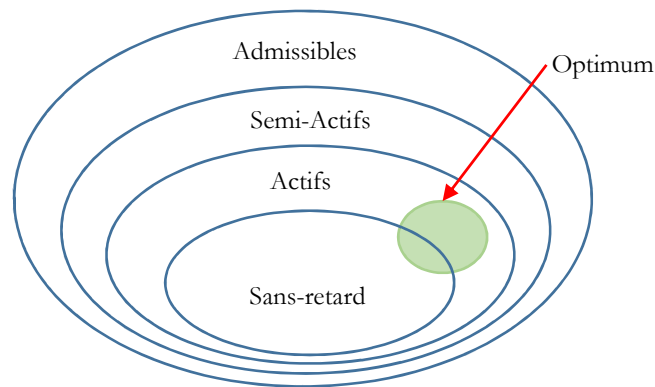


Figure (1.17) : Diagramme de Venn

Ainsi, les ordonnancements sans retard sont inclus dans le sous-ensemble des ordonnancements actifs ; qui sont eux-mêmes inclus dans le sous-ensemble des ordonnancements semi-actifs.

### 6.1. Ordonnancement admissible

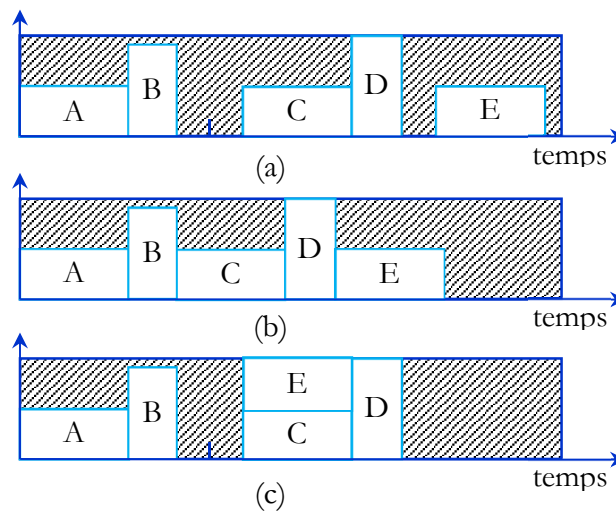
Un ordonnancement est dit admissible, si l'ordre d'affectation de ces tâches respecte toutes les contraintes du problème : les dates limitées, les contraintes de précédence entre les différentes tâches ainsi que la capacité maximale de la machine à tout instant.

Dans certains cas, des décalages à gauche sur certaines opérations sont nécessaires. Selon que l'ordre des opérations reste inchangé ou non, on distingue deux cas [Pinedo, 1995] :

- Décalage à gauche local : On parle de glissement à gauche, local lorsque l'avancement du début d'une opération ne remet pas en cause l'ordre des autres opérations,
- Décalage à gauche global : On parle de glissement à gauche global, lorsque l'avancement du début d'une opération engendre une modification au niveau de l'ordre relatif, au minimum de deux opérations.

Par exemple. Soit à ordonnancer cinq opérations A, B, C, D et E, telles que l'on doive respecter les deux contraintes de précédence suivantes :  $A < B < C$  et  $D < E$ , Un ordonnancement admissible est représenté Figure (1.18) (a). La solution pourrait être améliorée du point de vue du temps total d'exécution, en faisant :

- 1) Un glissement local de certaines tâches vers la gauche C et D de deux unités, puis E de trois unités Figure (1.18) (b).
- 2) Un repositionnement de la tâche E juste au-dessus de la tâche C est admissible mais change la relation  $D < E \rightarrow E < D$ , Figure (1.18) (c).



## 6.2. Ordonnements semi-actifs

Un ordonnancement est dit semi-actif s'il est valide et s'il est impossible de translater une opération vers la gauche pour obtenir un nouvel ordonnancement valide tout en conservant le même ordre de succession des opérations sur les différentes ressources. Autrement dit, dans un ordonnancement semi-actif, l'exécution de chaque opération commence aussi tôt que possible.

En présence d'écart maximum, cette définition n'est plus tout à fait exacte. Toute l'ambiguïté repose sur la phrase « s'il est impossible de translater une opération vers la gauche ». Considérons l'exemple suivant :

- deux ressources ( $M_1, M_2$ ) et un travail comportant deux opérations ( $O_{11}, O_{21}$ ).
- $O_{11}$  s'exécute sur  $M_1$ ,  $O_{21}$  sur  $M_2$ .
- $P_{11} = P_{21} = 1$ .



- $d(O_{21}, O_{11}) = -1$  (pas de temps mort autorisé entre la fin de  $O_{11}$  et le début de  $O_{21}$ , symbolisé par le lien vertical gras sur la Figure (1.18);
- $r_{11} = 0$ .

La solution valide de la Figure (1.19) n'est cependant pas optimale, il est possible de translater les deux opérations vers la gauche pour caler  $O_{11}$  à l'origine. Pourtant, individuellement, il est impossible de translater les opérations,  $O_{11}$  étant liée à  $O_{21}$  par une contrainte d'écart maximal. La solution proposée est semi-active au sens de la définition précédente. Il convient donc de modifier cette définition et de la reformuler de la manière suivante :

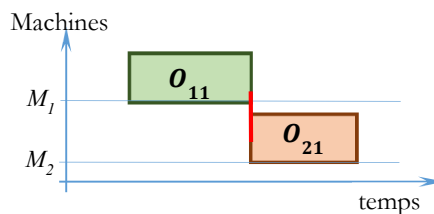


Figure (1.19) : Solution valide sans attente

### 6.3. Ordonnements actifs

Un ordonnancement est dit actif s'il est impossible d'avancer le début d'exécution d'une opération sans retarder une autre tâche ou violer une contrainte. Il est dit, actif généralisée, lorsqu'il est impossible de translater un sous-ensemble d'opérations vers la gauche, pour obtenir un nouvel ordonnancement valide, sans en retarder un autre.

Initialiser la date de disponibilité des machines à 0 ;

T : ensemble des opérations à ordonnancer ;

**Tant que** ( $T \neq \emptyset$ ) **faire**

En considérant les opérations de T dont toutes les opérations qui précèdent dans le même travail sont déjà placées.

Trouver une opération qui si on la place au plus tôt sur sa machine d'exécution  $M^*$ , aura la plus petite date de fin  $C^*$  ;

Parmi les opérations utilisant la machine  $M^*$  est dont toutes les opérations qui précèdent dans la même tâche sont déjà placées : se limiter aux opérations dont la date de début peut être strictement inférieure à  $C^*$  et parmi ces opérations, choisir la plus prioritaire (selon les règles de choix intégrées) noté  $O^*$  ;

Placer l'opération  $O^*$  sur la machine  $M^*$  et l'enlever de T ;

**Fin tant que**

Algorithme (1.1) : Générateur d'ordonnancement Actif

#### 6.4. Ordonnements sans délais

Un ordonnancement valide est Sans délai ou sans retard (non-delay), si et seulement si aucune opération n'est mise en attente, alors, aucune ressource ne reste inoccupée [T'Kindt et al. 2006]. Il est à noter que la transformation en un ordonnancement sans délai peut mener à une mauvaise solution plus du point de vue Makespan [Lopez et al. 1999]

[Baker, 1974] a énoncé une propriété qui stipule que, l'ensemble des ordonnancements semi-actifs est dominant dans les problèmes d'optimisation d'un critère régulier et le sous-ensemble des ordonnancements actifs est le plus petit ensemble dominant. Un générateur d'ordonnancement sans délais fonctionne de la manière suivante :

```

M : nombre de machines ;
T : tableau de dimension M ;
 $T(k) = 0 \quad \forall M_k \in M$  ;
T : ensemble de toutes les opérations de toutes les tâches ;
Tant que (T ≠ ∅) faire
    Chercher une machine k* disponible au plus tôt  $\min\{t(k)\}$  ;
    Chercher l'ensemble R des opérations qui peuvent être ordonnancées sur
    k* à l'instant t (k*) ;
    Si (R ≠ ∅) Alors
        |  $t(k^*) \leftarrow t(k^*) + 1$  ;
    Sinon
        | Choisir O* l'opération la plus prioritaire de R ;
        | Placer l'opération O* sur la machine k* ;
        | Enlever O* de T ;
    Fin si
Fin tant que
    
```

Algorithme (1.2) : Générateur d'ordonnancement Sans Délais

#### 6.5. Ordonnement statique/dynamique

Un problème d'ordonnancement est statique lorsque les choix d'ordonnancement sont faits avant l'exécution de l'application, et ne sont pas remis en cause durant cette exécution. C'est-à-dire l'algorithme d'ordonnancement a une connaissance à priori et complète de l'ensemble des tâches et de leurs caractéristiques comme échéances, durées d'exécution, contraintes de précédence et temps de déclenchement. L'analyse statique conduit à la construction calendrier donnant les dates de début et de fin d'exécution pour l'ensemble des tâches. L'algorithme d'ordonnancement statique opère sur l'ensemble de tâches et produit un schéma d'ordonnancement fixe et invariable. Cette approche n'est valide que pour des systèmes dont on sait qu'ils ne subiront aucune modification et aucune évolution.

Dans le cas contraire, les techniques statiques ne sont pas performantes dans la mesure où elles ne sont pas capables de remettre en cause les choix initiaux afin de les adapter en fonction de la réalité instantanée du système. Les approches dynamiques ou adaptatives permettent de faire face à différentes formes d'incertitude : durées d'exécution ou périodes d'activation inconnues, et de nouvelles tâches qui peuvent être activées de façon imprévue en cours d'exécution. Ainsi, l'ordonnancement varie au cours du temps, [Combaz, 2006].

Dans ce cas, aucune garanti n'est donnée quant au respect des contraintes de temps. Lorsqu'une nouvelle tâche survient, l'ordonnanceur fait une étude de faisabilité afin de déterminer s'il peut garantir l'ordonnancement correct de l'ensemble. Si oui, il établit un schéma d'ordonnancement et insère la tâche dans l'ensemble des tâches garanties. En cas d'échec, différentes alternatives restent possibles en fonction du temps disponible.

## 7. Problème d'ordonnancement Job Shop (JSSP)

Les problèmes de job shop se rencontrent, dans des entreprises manufacturières agroalimentaires par exemple. Ces entreprises fabriquent, plusieurs produits hystérogènes. Chaque produit est caractérisé par une gamme de fabrication spécifique, connue à l'avance [Egïese, 1990].

### 7.1. Description du problème d'ordonnancement job shop

L'ordonnancement d'atelier consiste à optimiser l'exploitation des moyens de production limités (machines), pour réaliser un ensemble varié de produits (lots). La complexité ne réside pas dans le processus de fabrication prédéterminé mais plutôt dans la combinatoire qui naît de la prise en compte des limitations de ressources.

Dans un problème d'atelier, un produit doit être fabriqué sur différentes machines. Chaque machine est une ressource disjonctive, c'est-à-dire qu'elle ne peut exécuter qu'une tâche à la fois, et les tâches sont liées exclusivement par des contraintes d'enchaînement. Plus précisément, les tâches sont regroupées en  $n$  entités appelées travaux ou lots. Chaque lot est constitué de  $m$  tâches à exécuter sur  $m$  machines distinctes.

L'ordonnancement job shop est associé à des lignes de production dédiées à la production de moyenne et de petites séries où les changements de produits sont fréquents (généralement des produits de caractéristique différentes). Le job shop est considéré comme étant un atelier à cheminement multiple. Chaque travail passe obligatoirement sur les machines selon une gamme fixée au préalable et spécifique à chaque travail. On parle, d'un atelier à flot multidirectionnel. Dans le cas où une opération nécessite l'exécution sur plus d'une machine on parlera d'un job shop à

machines dupliquées, et si les opérations peuvent passer plus d'une fois sur une machine il s'agira donc ce cas d'un ordonnancement réentrant.

Dans cette thèse, nous nous focaliserons en particulier sur le problème d'atelier à cheminements multiple (Job Shop) sans temps d'attente ni réentrant.

Soit un problème d'ordonnancement job shop. Les notations suivantes sont introduites pour décrire le problème. Soit  $M$  l'ensemble des  $m$  machines  $\{M_k\}_{k=1}^{k=m}$  et  $J$  ensemble des  $n$  job  $\{J_i\}_{i=1}^{i=n}$ . les jobs sont décrits par des gammes. Une gamme définit l'ordre, la durée et la séquence des machines où sont traitées les opérations d'un job. La gamme définit pour chaque job  $J_i$  constitue une suite de  $n_i$  opérations noté  $O_{ij}$  tel que  $J_i = \{O_{ij}\}_{j=1}^{j=n_i}$  et  $i = \{1, 2, \dots, n\}$ . L'opération  $O_{ij}$  constitue la  $j^{\text{ème}}$  opération du  $i^{\text{ème}}$  job. Un job ne peut commencer le traitement d'une opération que quand l'opération précédente dans la gamme est terminée.

Une opération  $O_{ij}$  doit être réalisée pendant un temps  $p_{ij}$  sur une machine  $M_k$ . cette opération doit être réalisée sans préemption.

On note  $\Phi_k$  l'ensemble des opérations devant être traitées sur la machine  $M_k$ . le nombre d'opération à traiter sur la machine  $M_k$  est donc  $e_k = |\Phi_k|$ . Une machine ne peut traiter qu'une seule opération à la fois. Les opérations de  $\Phi_k$  doivent donc être ordonnées, on dit qu'il y a disjonction entre elles car les intervalles de temps pendant lesquels la machine est occupée sont disjoints.

Plusieurs hypothèses sont souvent retenues lors de la résolution d'un problème d'ordonnancement Job Shop sont, les plus utilisées sont :

Le problème d'ordonnancement de type job shop présente des hypothèses à respecter et des conditions particulières sur l'environnement d'exécution concernant à la fois les ressources et les contraintes de temps. Ces hypothèses sont les suivantes :

- Tous les jobs sont disponibles dès le début de l'ordonnancement,
- Les machines ne sont pas identiques : chaque machine est unique. Lors de l'affectation, aucune choix de machines n'est posé,
- Les opérations d'un même job s'exécutent séparément sur les machines en respectant l'ordre des séquences  $O_i$  fixé par la gamme de production.
- La préemption des opérations n'est pas permise : une fois commencée, l'exécution d'une opération ne peut pas être interrompue,

- Toutes les données du job shop sont connues à l'avance : il s'agit alors du job shop statique (en particulier, aucun nouveau job n'est introduit après le début de l'ordonnancement).
- Les machines ont une disponibilité infinie. Elles sont supposées fiables et ne subissent aucune panne pouvant perturber ou arrêter l'exécution des opérations. C'est-à-dire, les machines sont disponibles sur tout l'horizon d'ordonnancement. Cette contrainte sera relaxée ultérieurement dans cette thèse.
- Les problèmes job shop classiques, appelés aussi carrés, traitent généralement des jobs ayant un nombre d'opérations égal au nombre de machines. Ainsi, toutes les opérations d'un même job s'exécutent sur des machines différentes.
- Les produits peuvent attendre dans les stocks de capacité illimitée.
- Les temps de traitement  $p_i$  sont déterministes et connus à l'avance,
- Le temps de montage et démontage sont inclus dans les temps de traitement,
- Les temps de transport sont négligeables.

### 8. Complexité des problèmes d'ordonnancement

Les problèmes d'ordonnancement sont des problèmes d'optimisation combinatoire, dont la complexité est démontrée pour la plupart d'entre eux. La théorie de la complexité classe les problèmes combinatoires en plusieurs catégories et donne une orientation sur le traitement de ces problèmes. Parmi ces classes, on distingue les problèmes de décision et les problèmes d'optimisation. Chaque problème d'optimisation peut être décrit sous forme d'un problème de décision, qui n'est pas nécessairement plus facile que le problème d'optimisation. Ainsi, il est possible d'analyser la difficulté d'un problème d'optimisation en considérant sa version de décision.

L'efficacité d'un algorithme est mesurée par son temps d'exécution, c'est-à-dire, pour toute instance des données en entrée de l'algorithme, combien d'opérations élémentaires nécessaires pour atteindre le résultat, cette mesure est aussi dite complexité algorithmique. Quand ce nombre d'opérations est borné par un polynôme, dépendant de la taille des entrées, alors l'algorithme est dit polynomial, par contre, si ce nombre d'opérations est borné inférieurement par une forme exponentielle de la taille des données, alors l'algorithme est dit exponentiel. Pour plus d'informations sur la théorie de la complexité, le lecteur peut consulter [cook, 1971], [Garey et al. 1979] et [Johnson, 2012]. Le but de la théorie de la complexité est la classification des problèmes de décision [Cook, 1971] suivant leur degré de difficulté de résolution.

La connaissance de la classe de complexité d'un problème combinatoire est décisive quant à la réponse algorithmique à un problème de décision ou d'optimisation. Trois classes sont évoquées dans ce qui suit à savoir la classe P, la classe NP et la classe NP-Complet.

La classe P contient tous les problèmes de décision solubles en temps polynomial, c'est-à-dire, les problèmes de décision pour lesquels il existe un algorithme polynomial permettant de répondre à la question par un oui, c'est-à-dire en un temps  $O(n^k)$  avec  $n$  la taille du problème traité,  $k$  un entier naturel et une constante indépendante de  $n$ .

On appelle cette classe celle des problèmes dits faciles.

La classe NP est une extension de la classe P. NP est l'abréviation pour « Non deterministic Polynomial time ». Cette classe renferme tous les problèmes de décision dont on peut associer à chacun d'eux un ensemble de solutions potentielles (de cardinal au pire exponentiel) tel qu'on puisse vérifier en un temps polynomial si une solution potentielle satisfait la question posée. Le terme non-déterministe désigne un pouvoir qu'on incorpore à un algorithme pour qu'il puisse deviner la bonne solution. Pour définir la classe NP-Complet, il est intéressant de définir la notion de transformation polynomiale.

Transformation polynomiale : Une transformation d'un problème  $A_1$  en problème  $A_2$  est dite polynomiale, s'il existe un algorithme polynomial qui transforme n'importe quelle instance  $I_1$  du problème  $A_1$  à une instance  $I_2$  du problème  $A_2$ , telle que la réponse à l'instance  $I_1$  est « oui » si et seulement si la réponse à l'instance  $I_2$  est « oui ». Cette transformation est aussi appelée réduction.

La classe NP-Complet : Un problème  $A$  est dit dans la classe NP-Complet, si  $A$  appartient à la classe NP et tous les problèmes de la classe NP peuvent être transformés en problème  $A$ . Comme la transformation polynomiale est transitive, il suffit de chercher un seul problème  $B$  de la classe NP qui peut être réduit polynomialement au problème  $A$ , afin que ce dernier devienne NP-Complet.

En fin ; il est intéressant de savoir comment exploiter ces différentes notions pour l'analyse des problèmes combinatoires en général, et les problèmes d'ordonnancement en particulier. En effet, pour un problème d'optimisation, il suffit d'analyser le problème de décision équivalent. Si ce problème de décision appartient à la classe P, alors le problème d'optimisation est polynomial, si le problème de décision est NP-Complet, alors on dit que le problème d'optimisation est NP-Difficile. L'intérêt de l'analyse de la complexité est de savoir, si un problème d'optimisation appartient à la classe P alors il est soluble en temps polynomial, et si on prouve qu'un problème d'optimisation appartient à la classe NP-Difficile, alors il en résulte une explosion combinatoire, de l'ordre de l'exponentielle, lors de la recherche de la solution optimale, [Jihène, 2004], [Ammar, 2009].

En conclusion, les méthodes exactes sont souvent utilisées pour des problèmes de petite taille, en revanche les méthodes de résolution approchées doivent être utilisées pour les problèmes de grande taille, comme les heuristiques, les algorithmes d'approximation, les métaheuristiques, etc.

La plupart des problèmes d'ordonnancement se classe dans la catégorie des problèmes dits NP-difficiles [Garey et al. 1979]. Pour ces problèmes, il n'existe pas d'algorithmes de complexité optimale pour les résoudre à l'optimalité. Dans un problème à machine unique, par exemple, comportant  $n$  travaux, il y a  $n!$  solutions possibles. Plus récemment, dans son ouvrage, [Pinedo, 2002] donne la complexité de plusieurs problèmes d'ordonnancement industriel. Pour résoudre de tels problèmes, [Blazewicz, 1994] suggèrent trois avenues : utiliser des méthodes exactes pour de petites instances, utiliser des heuristiques et utiliser des méthodes hybrides.

### 8.1. Complexité du problème d'ordonnancement job shop

Les problèmes de job shop sont en général NP-complets, même si l'atelier est simple. En effet, [Lenstra et al. 1977] ont montré que les ateliers possédant plus de trois machines, ou un nombre de tâches supérieur ou égal à trois, sont NP-difficiles même si la préemption est permise. De même, pour les problèmes à deux machines, dès qu'il y a recirculation, ils deviennent fortement NP-difficiles. L'expérience a montré que les problèmes de job shop ayant un nombre  $m > 2$  machines, et optimisant les critères  $C_{max}$  et  $F$  sont NP-difficile au sens fort [Jen, [2001], même en utilisant des heuristiques. Théoriquement, le problème du job shop  $J//C_{max}$  peut se réduire polynomialement aux problèmes  $J//T_{max}$  et  $J//L_{max}$ . De même, le problème  $J//F$  peut se réduire aux problèmes  $J//T$  et  $J//L$ . par conséquent, ces quatre dernières instances du job shop sont NP-difficile au sens fort. En effet, une instance de job shop de taille  $10 \times 10$  proposée par [Muth et al. 1963] n'est optimalement résolue qu'en 1989 par [Carlier et al. 1989].

Pour résoudre un problème d'ordonnancement de manière efficace, il faut prendre en compte la particularité de chaque problème. C'est pourquoi il faut s'assurer de la classe de complexité associé au problème à ordonnancer. Il existe plusieurs méthodes de résolution du problème d'ordonnancement, la connaissance de la classe de complexité peut nous aider énormément quant au choix judicieux de la méthode de résolution. Dans le chapitre 4 nous passons en revue différentes approches exactes et approchées pour la résolution des problèmes d'ordonnancement.

### Conclusion

Dans ce chapitre, nous avons positionné l'ordonnancement dans le système de production, où l'accent est mis sur son importance cruciale dans le processus de production, et sa participation incontournable dans la mise sur les marchés des produits à moindre coût et dans de courts délais. On a introduit les principales caractéristiques d'un problème d'ordonnancement, pour s'intéresser ensuite à la complexité des problèmes qui peuvent être rencontrés.

Dans la plupart des recherches existantes dans la littérature portant sur l'ordonnancement, et particulièrement des ateliers à cheminement multiple (job shop), on considère les ressources (machines) comme disponibles à tout moment ou éventuellement durant l'horizon d'ordonnancement. La fonction maintenance qui veille sur la disponibilité des machines d'une entreprise n'est jamais prioritaire sur la fonction production, pour effectuer des interventions préventives. Pour mettre en exergue ce problème, nous verrons dans le chapitre suivant l'importance de la maintenance dans l'entreprise et de son intégration dans l'ordonnancement de la production.



---

## **CHAPITRE 2.**

# **ORDONNANCEMENT DE LA MAINTENANCE DANS UN SYSTÈME DE PRODUCTION**

# Chapitre 2.

## Ordonnancement de la maintenance et de la production.

---

***Résumé :** Ce chapitre vise à décrire en premier lieu les concepts de base de la maintenance et son importance dans un système de production. En second lieu, il traite l'interaction entre ces deux fonctions qui exploitent la même ressource machine. Cette interaction génère des conflits ayant pour conséquence finale une perte de disponibilité et une augmentation des coûts.*

*Pour une cohérence optimale entre les fonctions production et maintenance, nous faisons recours au concept d'intégration et notamment l'ordonnancement intégré.*

---

## Introduction

La fonction production est considérée comme la fonction mère autour de laquelle gravite les autres fonctions d'aide et de soutien, essentiellement la fonction maintenance.

La maintenance est une fonction stratégique, considérée comme fonction de soutien à la production dans l'entreprise industrielle, sa vocation est d'assurer le bon fonctionnement de l'outil de production. Actuellement, avec le développement, la complexité croissante des machines, la concurrence acharnée et la clientèle de plus en plus exigeante, les industries pour gagner ces défis, sont contraintes de progresser continuellement. Elles doivent fournir des produits de haute qualité, des livraisons fiables et rapides, tout cela avec de coûts raisonnables. Dès lors, l'objectif principale de la maintenance est d'assurer la pérennité des équipements, de diminuer les pannes et de réduire les coûts de révision et de remise en état de fonctionnement.

Les fonctions maintenance et production sont en étroite collaboration, ayant une interface commune qui est la ressource « machine ». Cette interaction peut générer des conflits, nécessitant ainsi un décloisonnement en vue d'un renforcement des liens et d'une synchronisation des tâches. Cette manière d'agir se définit par la notion d'intégration visant la cohérence entre ces deux fonctions.

Malgré les avancées significatives des technologies de l'information, l'intégration d'entreprise reste un défi à relever pour les organisations, aussi bien au niveau local par l'intégration intra-entreprise qu'au niveau global par celle inter-entreprises. La même difficulté apparaît en ce qui concerne la capacité de ces organisations à créer une synergie entre les différentes fonctions de l'entreprise à travers des actions d'organisation notamment la planification et l'ordonnancement, ceci pour contribuer à la satisfaction des besoins clients. L'un des freins à cette intégration est l'absence d'une réelle stratégie. Il s'avère donc nécessaire d'élaborer cette stratégie, en tenant compte de l'environnement interne et externe à l'entreprise, et de définir une méthodologie de planification globale des tâches des différentes fonctions de l'entreprise, essentiellement les fonctions production et maintenance objet de notre recherche.

Dans la plupart des travaux dédiés à l'ordonnancement de production, les ressources sont supposées être disponibles en continu pour effectuer des jobs. Ceci n'est pas toujours vrai : les différentes ressources aussi bien matérielles qu'humaines peuvent être indisponibles pour diverses raisons. Les dates et les durées des périodes d'indisponibilité sont connues dans certains cas : congés du personnel, activités de maintenance programmée, etc. D'autres périodes d'indisponibilité telles que les pannes machines ou les défections du personnel ne sont pas prévisibles. La présence des "trous" dans un planning influence les processus de production de façon significative. Le problème étant

qu'une ressource additionnelle pour absorber cette charge de travail n'est pas forcément disponible. Il est alors nécessaire de trouver la meilleure façon de répartir la charge de travail entre les machines en prenant en compte ces périodes d'indisponibilité, le type des opérations que les machines peuvent effectuer, et l'ordre entre ces opérations. La meilleure façon d'appréhender cette problématique et de faire recours aux techniques d'ordonnements.

Ce chapitre s'articule autour des deux fonctions qui sont la production et la maintenance à travers en premier lieu des concepts et des définitions de la maintenance, les lois usuelles utilisées en fiabilité, notamment la loi de Weibull. En second lieu la notion d'intégration des fonctions en question et les effets conflictuels qui peuvent exister sont abordés en s'appuyant sur l'importance de l'ordonnement intégré pour gérer ce type de conflit.

## **1. Évolution de la maintenance**

La fonction maintenance est, par nature, une activité transversale de l'entreprise au service des autres fonctions dont elle renforce l'efficacité. Les années cinquante et soixante, ont été marquées par l'apparitions des premières approches scientifiques de la gestion de maintenance [Pierskalla et al. 1976], [McCall, 1965]. À cette époque, son objectif a été, la remise en service le plus rapidement possible de l'équipement défaillant sans se soucier de faire une analyse préalable des causes et conséquences des défaillances. [Monchy, 2000]. En effet, elle était rarement considérée comme une activité stratégique au sein de l'entreprise. Dans la même période, les premiers modèles de recherche opérationnelle visant l'optimisation des programmes de la maintenance sont apparus [Verdol, 1988]. Dans les années soixante-dix, grâce aux contrôles des ateliers et à la surveillance, l'utilisation de l'information sur l'état actuel de l'équipement a permis de se concentrer sur des techniques pouvant prédire des défaillances. Cela semblait être plus efficace que les gros programmes de maintenance préventive. Des études détaillées de la part des fabricants, des défaillances de leurs produits ont abouti à de meilleures conceptions, avec moins de défaillances. [Benbouzid, 2005]. Dans les années quatre-vingt, l'ordinateur apporte de l'aide à la fonction maintenance. Initialement, il a été utilisé pour faciliter les tâches administratives, ensuite pour la gestion de l'information disponible. À nos jours, l'utilisation de l'outil informatique a contribué à la prise de décision, [Boulenger 1988]).

John Moubray dans son livre *Reliability Centered Maintenance* a résumé l'évolution de la maintenance depuis l'années 1940, jusqu'au 2000 en trois générations, la première concerne le Remplacement du composant lorsqu'il est détruit, la seconde est caractérisée par : le processus jouit d'une disponibilité supérieure, d'une durée de vie des composants plus longue et de réflexion sur l'optimisation des coûts. Tant disque dans la troisième l'objectif est d'améliorer la disponibilité et la

fiabilité dans une plus grande sécurité, d'assurer une meilleure qualité des produits, de trouver les solutions aux problèmes néfastes à l'environnement et en fin garantir un meilleur rapport coût efficacité, [Moubray, 2000], Figure (2.1).

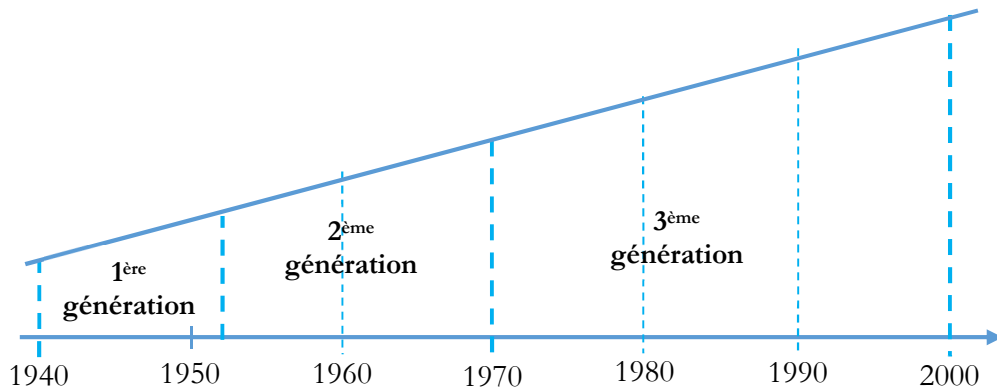


Figure (2.1) : Évolution de la maintenance depuis 1940, [Vrignat et al. 2012]

## 2. Définition de la maintenance

L'association française de normalisation [AFNOR 2001] définit la maintenance comme étant l'ensemble de toutes les actions administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise. Il est précisé dans cette définition que des événements peuvent survenir sur tout bien et entraîner des conséquences sur son fonctionnement.

À travers le terme de défaillance, il y a la notion d'un événement anormal survenu à un instant donné de fonctionnement de l'équipement. Cela le diffère de la dégradation, qui est une évolution de l'état de fonctionnement de l'équipement dans le temps. Cette évolution provoque l'éloignement de ce que l'on obtient en fonctionnement en mode dégradé par rapport à ce que l'on devrait obtenir au cours d'un fonctionnement normal ou en mode de fonctionnement nominal. Si la dégradation continue dans le temps et l'équipement n'est pas pris en charge, nous nous retrouvons dans l'état de panne.

## 3. Place de la maintenance dans le système de production

L'évolution et la complexité grandissante des systèmes de production, sous des contraintes de plus en plus pressantes, d'un marché concurrentiel, des exigences client, d'un coût de revient croissant ainsi que le besoin de produire dans des courts délais, ont obligé les industriels à créer de nouveaux concepts d'organisation et de nouvelles manières d'intervenir sur des structures de production.

En effet, l'objectif de la maintenance est de tout mettre en œuvre pour maintenir le système de production en état de fonctionnement et ceci en évitant les pannes, sinon en agissant rapidement lorsqu'elles surviennent, pour augmenter la disponibilité de l'outil de production.

Les concepts de la maintenance définissent des façons de faire pour maximiser la performance globale de l'entreprise. La mise en œuvre de ces concepts exige des ressources humaines compétentes, des outils et du matériel adaptés aux équipements et aux installations à entretenir, un système de gestion de pièces de rechange adéquat et un système d'information bien pensé pour assurer un échange efficace entre les différents intervenants.

Nous allons développer dans ce qui suit, la définition et concepts de base liée à la maintenance.

#### **4. Types de maintenance**

En se référant à la définition de la maintenance, nous apercevons deux mots clés : maintenir et rétablir. Le premier fait référence à une intervention préventive, avant la survenue de la défaillance, tandis que, le deuxième désigne une activité corrective qui réagit juste après l'occurrence de la panne [Barlow et al. 1960], [Valdez et al., 1989]. En effet, la norme AFNOR distingue plusieurs politiques de maintenance : la maintenance préventive (MP) et la maintenance corrective (MC).

La maintenance préventive peut être systématique, conditionnelle ou encore prévisionnelle, tandis que la maintenance corrective peut être palliative ou curative. Bien que les deux stratégies de maintenance évoquées soient les principales, on rencontre aussi la maintenance améliorative qui permet d'améliorer les fonctions des équipements. La Figure (2.2) résume les différents concepts et indicateurs de maintenance qu'on trouve dans la littérature, en indiquant les événements initiateurs.

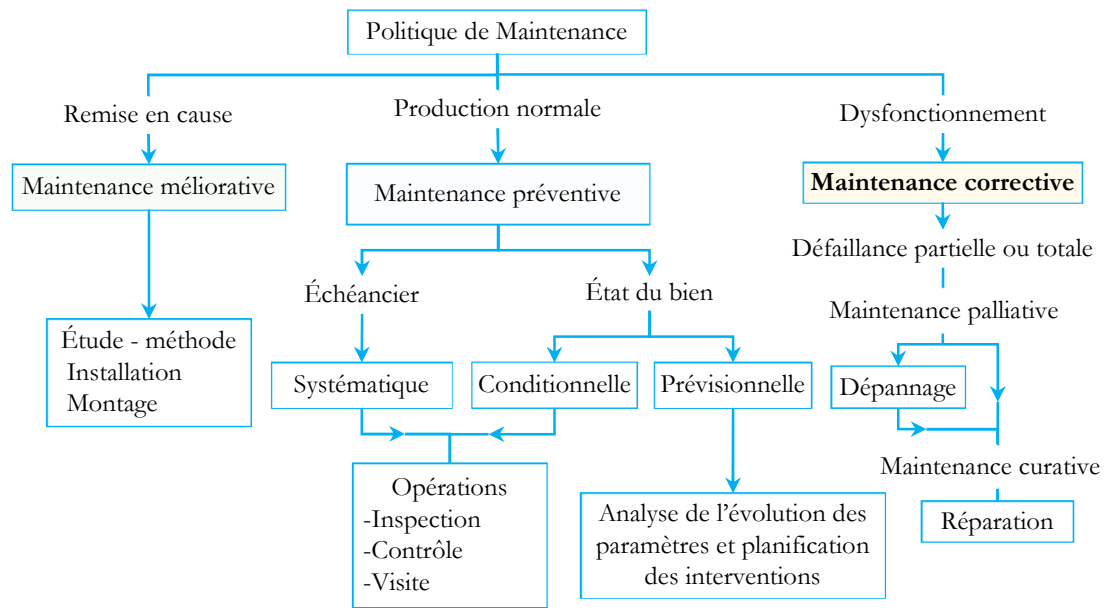


Figure (2.2) : Politiques de maintenance et évènements révélateurs

Avant de détailler chacune des stratégies de maintenance, on présente dans le paragraphe suivant les principales activités de maintenance.

#### 4.1. Les activités de la maintenance

- **L'inspection** : C'est un contrôle de conformité réalisé en mesurant, observant, testant ou calibrant les caractéristiques significatives d'un bien.
- **La surveillance** : C'est l'activité exécutée manuellement ou automatiquement ayant pour objectif d'observer l'état réel d'un bien.
- **La réparation** : Ce sont les actions physiques exécutées pour rétablir la fonction requise d'un bien en panne.
- **Le dépannage** : Ce sont les actions physiques exécutées pour permettre à un bien en panne d'accomplir sa fonction requise pendant une durée limitée.
- **L'amélioration** : Ensemble des mesures techniques, administratives et de gestion, destinées à améliorer la sûreté de fonctionnement d'un bien sans changer sa fonction requise.
- **La modification** : Ensemble des mesures techniques, administratives et de gestion, destinées à changer la fonction d'un bien.
- **La révision** : Ensemble complet d'examens et d'actions réalisés afin de maintenir le niveau requis de disponibilité et de sécurité.
- **La reconstruction** : L'objectif de la reconstruction est normalement de donner à un bien une vie utile qui peut être plus longue que celle du bien d'origine.

## 4.2. La maintenance corrective

La norme française de normalisation AFNOR X60-010 définit la maintenance corrective comme étant un l'ensemble des activités réalisées après la défaillance du bien ou la dégradation de sa fonction pour lui permettre d'accomplir une fonction requise, au moins provisoirement.

Les défauts, pannes ou avaries diverses exigeant une maintenance corrective entraînent une indisponibilité immédiate ou à très brève échéance des matériels affectés et/ou une dépréciation en quantité et/ou qualité des services rendus. À partir d'un principe simple et intuitif, devant une telle situation, il faut réparer le matériel défectueux, ce qui sous-entend que s'il fonctionne, on n'y touche pas.

Les interventions effectuées dans le cadre d'une maintenance corrective ont pour objet la remise en état de fonctionnement des matériels ou équipements à la suite d'une défaillance totale ou partielle, d'altération ou cessation de l'aptitude d'un bien à accomplir la fonction requise. Ces interventions sont composées essentiellement, de la localisation de la défaillance partielle ou complète et son diagnostic, la remise en état avec ou sans modification, et enfin le contrôle du bon fonctionnement, Figure (2.3).

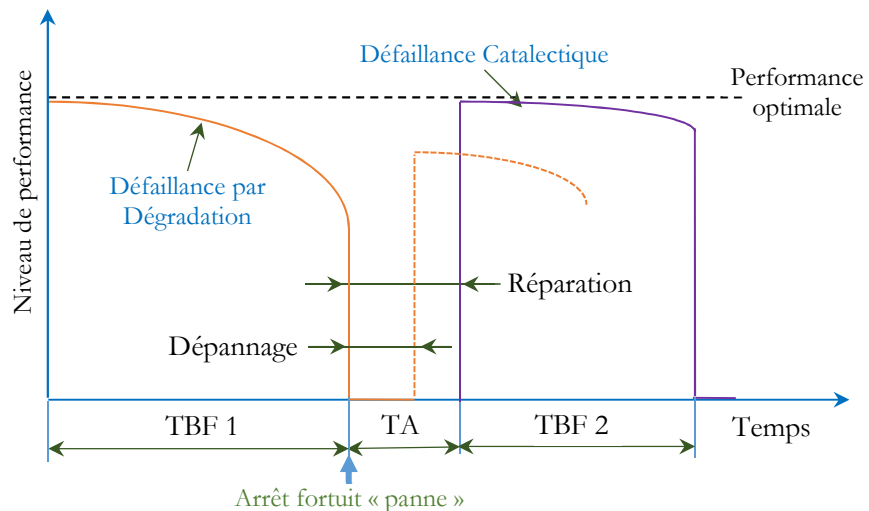


Figure (2.3) : Graphe caractéristique de la maintenance corrective

La maintenance corrective est réservée au matériel peu coûteux, non stratégique pour la production et dont la panne aurait peu d'influence sur la sécurité.

Selon la nature des opérations réalisées, on distingue généralement deux niveaux de maintenance. Dans le premier on parle de la maintenance palliative, lorsque, l'intervention a un caractère provisoire dans le sens où elle nécessitera forcément une intervention ultérieure. Dans le second, une opération de maintenance curative se caractérise par la recherche des causes initiales de



la défaillance et par la réalisation des opérations visant à rendre le système opérationnel et ainsi éviter toute nouvelle occurrence de cette défaillance.

### **4.3. La Maintenance Préventive.**

Par opposition à la maintenance corrective, la maintenance préventive regroupe les opérations de maintenance ayant pour objet de réduire la probabilité de défaillance et d'améliorer par conséquent, la disponibilité de l'outil de production. Ces opérations anticipent l'occurrence de toute défaillance qui risque d'entraver la production. Ce concept de maintenance est basé sur une inspection périodique de l'outil de production selon des critères prédéterminés, afin de juger de ce bon état de fonctionnement. C'est dans cet objectif que sont développées les stratégies de maintenance préventive. La maintenance préventive peut être de trois types : systématique, conditionnelle ou encore prévisionnelle.

La maintenance préventive s'adresse aux éléments provoquant une perte de production ou des coûts d'arrêts imprévisibles jugés importants pour l'entreprise. L'analyse de ces coûts met en évidence un gain important dû aux arrêts qu'elle permet d'éviter.

#### **4.3.1. La Maintenance Préventive systématique.**

La norme française de normalisation a proposé la définition suivante : la maintenance préventive systématique comprend l'ensemble des actions destinées à restaurer, en totalité ou partiellement, la marge de disponibilité des équipements non défaillants. Selon cette norme, la maintenance préventive systématique est exécutée à des intervalles de temps préétablis ou selon un nombre défini d'unités d'usage Figure (2.4) mais sans contrôle préalable de l'état de l'équipement, AFNOR (2001).

La maintenance préventive systématique est effectuée selon un échéancier établi, sur la base de critères d'usure tels que le MTTF ou MTBF et le MTTR. Ces derniers sont fixés sur la base de l'historique de l'équipement ou données par le constructeur. C'est pourquoi ce type de maintenance est appelé aussi maintenance préventive basée sur la durée de fonctionnement [Boulenger, 1988]. Les opérations de maintenance se traduisent par le remplacement systématique d'un certain nombre de composants identifiés par cet échéancier.

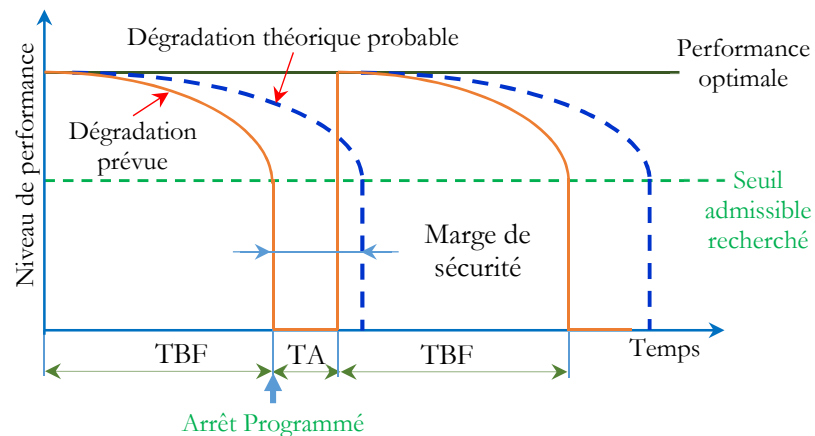


Figure (2.4) : Graphe caractéristique de la maintenance préventive systématique

La maintenance préventive systématique se traduit donc par deux types d'actions :

- Interventions planifiées qui consistent à nettoyer, réparer ou remplacer certains matériels tels que des composants ou sous-ensembles d'équipements,
- Inspections périodiques qui consistent à contrôler ces mêmes composants et sous-ensembles, d'effectuer des révisions, mineures ou majeures, d'équipements.

La périodicité d'intervention est déterminée à partir des préconisations du constructeur, suit à l'expérience acquise lors de la remise en service ou après une révision complète ou partielle de l'équipement, par l'exploitation fiabiliste réalisée à partir d'essais, d'historiques ou de résultats fournis par des visites préventives initiales; les lois de Weibull et exponentielles, ..., permettent de déterminer la MTBF associée à un intervalle de confiance, ou à l'issue d'une analyse prévisionnelle de fiabilité.

#### 4.3.2. Maintenance préventive conditionnelle

La maintenance préventive conditionnelle est une maintenance préventive subordonnée à un type d'événements prédéterminé Figure (2.5) par autodiagnostic, information donnée par un capteur, mesure d'une usure, etc., révélateur de l'état de dégradation d'un bien [Boulenger, 1988].

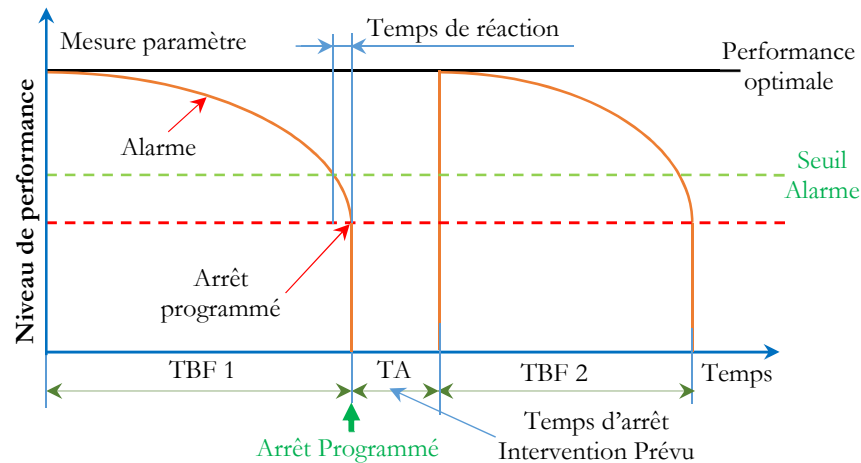


Figure (2.5) : Graphe caractéristique de la maintenance préventive conditionnelle.

Elle dépend donc, de l'expérience et fait intervenir des informations recueillies en temps réel [Boitel et al. 1990]. Elle sert à optimiser la maintenance en déclenchant les révisions au dernier moment et en limitant les travaux systématiques, donc les coûts, [Lavina, 1992]. La maintenance préventive conditionnelle est appelée aussi maintenance prédictive, ce terme est non normalisé.

#### 4.3.3. Maintenance préventive prévisionnelle

La maintenance prévisionnelle est subordonnée à l'analyse de l'évolution surveillée de la dégradation de l'équipement. Elle permet ainsi d'optimiser le planning de maintenance, en retardant ou en avançant les dates d'intervention.

La courbe d'évolution d'un défaut étant connue, il est possible d'en extrapoler sa tendance pour prévoir la date de défaillance. À partir de cette prévision, la date du diagnostic et du déclenchement de l'intervention de maintenance est planifiée afin que cette dernière soit terminée avant que le niveau requis pour le paramètre étudié ne soit dépassé, [Souris, 1990], Figure (2.6).

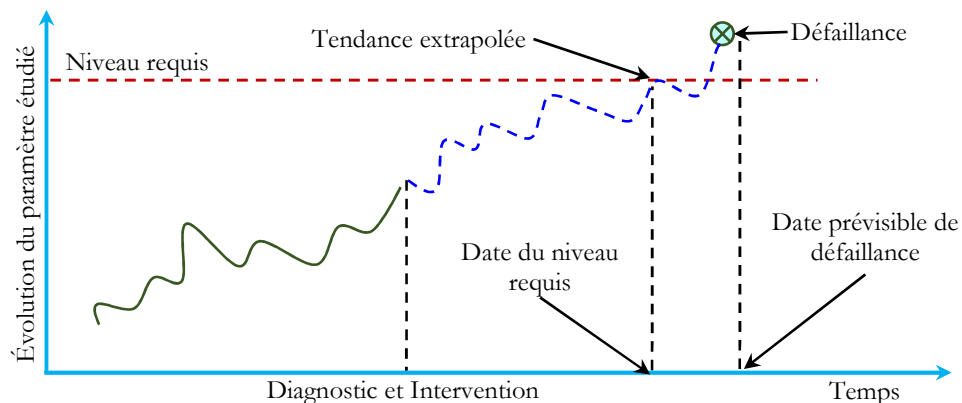


Figure (2.6) : Schématisation de la maintenance préventive prévisionnelle

## 5. Politique de maintenance

Le choix entre les méthodes de maintenance s'effectue dans le cadre de la stratégie de la maintenance et doit s'opérer en accord avec la direction de l'entreprise. Ce choix est fait dans le but de réduire la probabilité de défaillance d'un ou des équipements, qui subsiste inévitablement une part de maintenance corrective. Il est donc nécessaire de considérer des stratégies qui combinent les deux types de maintenance corrective et préventive.

La mise en place d'une politique de maintenance préventive nécessite une analyse rigoureuse du système de production, des modes de dégradation, des paramètres physiques pertinents, des moyens à mettre en œuvre, les coûts de maintenance et les coûts de perte de production, des objectifs en disponibilité et en gain économique, des qualifications et des réticences des personnels et des conséquences sur l'organisation générale du service.

Il faut choisir pour chaque machine ou groupe de machines un type de maintenance approprié. On peut utiliser des critères de choix simple, résumés par des arbres présentée sur la Figure (2.7).

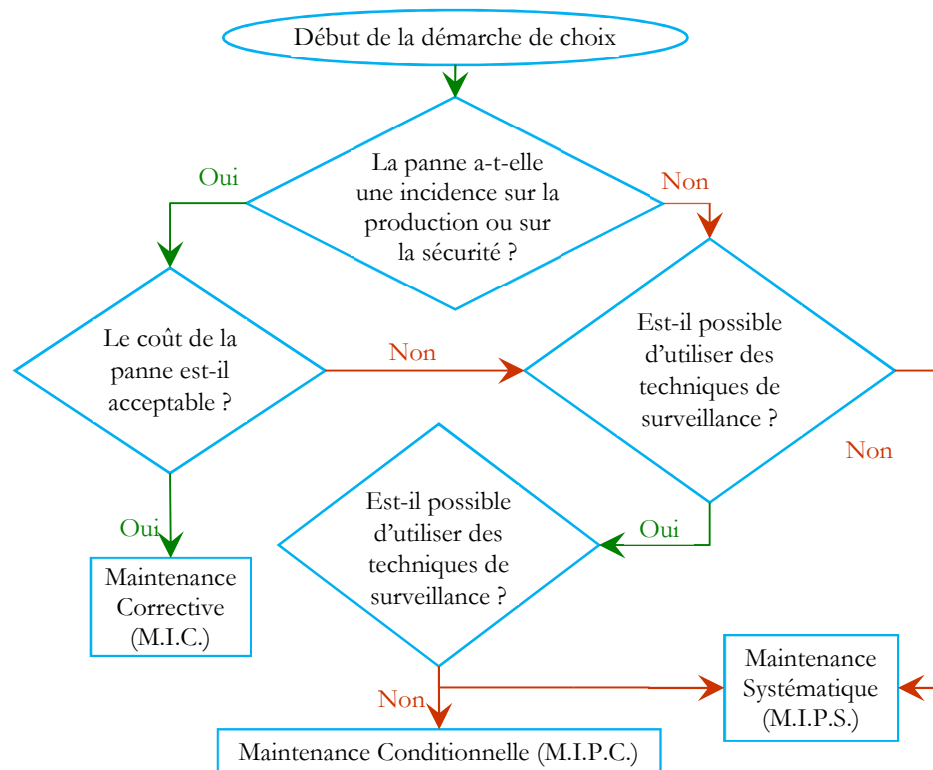


Figure (2.7) : Choix d'une politique de Maintenance

Le choix de type de maintenance peut être fait aussi en suivant deux cas :

- 1) Le premier cas : le type de maintenance est choisi en fonction de l'évolution du taux de défaillance de l'incidence sur la sécurité et la progressivité de la panne.

2) Le deuxième cas : le choix est fait, en estimant pour chaque élément, le rapport entre le coût, d'une intervention préventive effectuée dans de bonnes conditions  $C_b$ , et le coût supplémentaire supporté en cas de défaillance  $C_d$ . Si le rapport  $\frac{C_d}{C_b}$  est négligeable, la maintenance corrective est la plus intéressante. Si le rapport est important, une maintenance préventive systématique ou conditionnelle s'impose. Dans ce cas, si le composant présente des paramètres mesurables (huile dans une boîte à vitesse, éléments tournants, etc.), on peut choisir la maintenance conditionnelle, sinon la maintenance systématique est adoptée.

On peut alors classer les éléments de la machine en fonction de la valeur de ce rapport en : vital ( $\frac{C_d}{C_b}$  élevé), important ( $\frac{C_d}{C_b}$  moyen), secondaire ( $\frac{C_d}{C_b}$  faible) [Fleurquin et al. 2006].

Le niveau préventif à appliquer à un équipement relève d'un choix réfléchi dont nous étudierons les critères. Avec l'hypothèse, si le préventif est efficace, le temps passé à corriger les défaillances décroît, ainsi que les coûts associés au correctif résiduel.

La Figure (2.8) montre que les coûts directs et indirects de maintenance varient dans des sens opposés, la résultante des deux coûts est une courbe en baignoire et la courbe des coûts du préventif ou des coûts d'indisponibilité en fonction du temps passé à le mettre en œuvre est linéaire. Une maintenance préventive peu développée entraîne des coûts indirects, dus aux arrêts de production, trop importants. Inversement, trop de préventif entraîne des coûts directs, dus à des investissements humains et matériels, trop importants. Le coût du correctif ne tient compte que des coûts maintenance. S'il intègre le coût des pertes de production dues aux arrêts, et selon les entreprises, le coût sera souvent plus élevé et la zone dite niveau optimal de préventif sera décalée vers la droite ce qui suppose des taux de préventif plus importants. Ce qui prouve, que faire de la maintenance, ce n'est pas faire du préventif à tout prix. L'objectif est donc d'optimiser le plan de maintenance préventive en cherchant un équilibre entre les coûts directs et indirects et donc, entre la disponibilité des équipements et les coûts de maintenance.

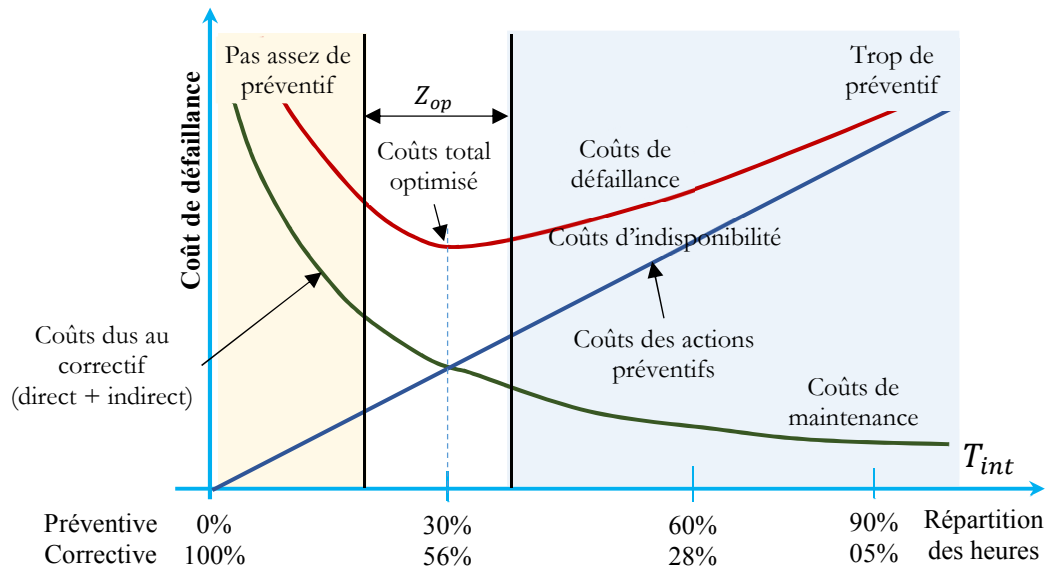


Figure (2.8) : Évolution des coûts de défaillance

$T_{int}$  : Cumule des temps d'indisponibilité en heurs par mois

$Z_{op}$  : Zone d'optimum économique

## 6. Les fonctions opérationnelles d'une structure maintenance.

La structure de la maintenance est constituée des trois principales fonctions opérationnelles : la fonction Méthodes, la fonction Ordonnancement et la fonction Réalisation. Figure (2.9).

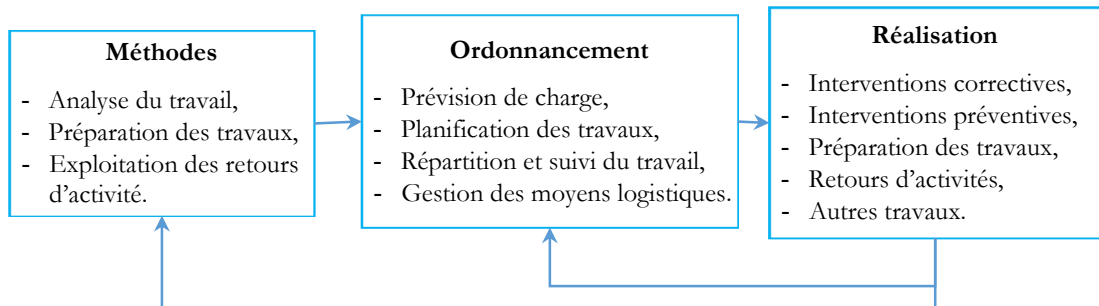


Figure (2.9) : Les trois fonctions opérationnelles de la maintenance, [Monchy, 2003]

### 6.1. La fonction étude de maintenance (méthodes)

La maintenance doit être orientée par les objectifs généraux de l'entreprise et se réguler à partir de l'écart constaté entre les performances attendues et celles qui sont observées.

La fonction études rassemble toutes les analyses effectuées à l'abri de la pression du quotidien pour définir une stratégie de maintenance en se fixant des objectifs mesurables. Elle inclut :

- l'analyse des besoins en fiabilité, maintenabilité et soutien logistique.

- l'analyse du retour d'expérience.
- l'élaboration du programme de maintenance préventive.
- la détermination des pièces de rechange et des matières à tenir en stock.
- les études de modifications de matériels.
- une participation à la définition du cahier des charges des investissements.

Chacune de ces analyses contribue à l'atteinte des performances attendues citées ci-dessus.

## **6.2.La fonction préparation des interventions**

Le rôle principal de la fonction préparation consiste à déterminer « comment on fait ». Le préparateur rédige des procédures (gammes) qui décrivent et mémorisent des opérations qui peuvent être répétitives.

Ces documents donnent le détail des interventions à réaliser sur les matériels en précisant :

- les ressources nécessaires (outillage, pièces de rechange, nombre et qualification des intervenants, services extérieurs, temps d'intervention...).
- les actions à entreprendre (consignations, préparation de chantier, procédures de sécurité, points d'arrêt et vérifications...).

Ce travail produit donc des données essentielles et il peut avoir une forte incidence sur les délais et sur les coûts.

## **6.3.La fonction d'ordonnancement.**

L'ordonnancement est l'une des fonctions opérationnelles de la maintenance, située entre la fonction méthode, chargée de la définition des tâches à effectuer et des moyens à mettre en œuvre, et la fonction réalisation chargée de leur exécution, [Monchy, 2003].

Le problème d'ordonnancement de la maintenance revient à un problème d'affectation de ressources de maintenance à la réalisation d'ordre de travail de maintenance (opération) sur des machines du système de production, les ressources de maintenance étant constituées des opérateurs de maintenance [Coudert, 2000]. Les missions principales de l'ordonnancement sont la planification des travaux, l'optimisation des moyens en fonction des délais et chemins critiques, et le contrôle de l'avancement des travaux. Cela peut s'exprimer sous la forme ; prévoir à l'instant (t) et un endroit (X) où un personnel (P) muni d'un outillage (O) et des matières (m) exécutera la tâche M en harmonie avec les autres travaux connexes, [Aghezzaf et al., 2008], [Halimi, 2014]. Malheureusement cette fonction stratégique est peu visible, et souvent peu étudiée. Par contre, son absence est fortement remarquable ; tâches préventives négligées, gaspillage de temps en recherche de moyens indispensables, améliorations toujours reportées à plus tard, techniciens parfois inoccupés associés à

des heures supplémentaires évitables, etc. La difficulté principale de l'ordonnancement vient du caractère fortuit de la panne : comment intégrer les dépannages à un planning [Vassiliadis et al., 2000]. Certains services de maintenance prétextent cette contradiction pour ne pas ordonnancer leurs activités, si 90% de leurs activités sont fortuites, ils ne peuvent guère faire autrement. Mais si grâce à une politique de prévention, ce taux passe à moins de 50% ce fortuit devient programmable [Deloux et al., 2007].

### **6.3.1. Particularité de l'ordonnancement de la maintenance**

L'ordonnancement des tâches de la maintenance est d'une étroite dépendance à la production. D'où, une bonne synchronisation est indispensable entre les services production et maintenance afin de profiter des arrêts de production pour réaliser des opérations de maintenance.

Deux particularité de l'ordonnancement de la maintenance sont à noter [Paz, 1994], la première est la variété et l'incertitude des travaux de maintenance ; la durée d'une intervention ne peut pas être connue de manière sûre tant que l'on n'a pas examiné l'équipement, la deuxième est la date de réalisation d'une tâche de maintenance préventive, qui peut être avancée ou retardée par rapport à la date optimale ou à l'intervalle optimal de visite [Gits, 1994].

### **6.4. Les objectifs de l'ordonnancement de la maintenance**

Les objectifs de l'ordonnancement de la maintenance sont fixés sous condition de la satisfaction d'un nombre important de contraintes hétérogènes. On peut citer entre autres, la dépendance de l'ordonnancement de la maintenance à la production, arrêts de fabrication, taux d'utilisation des équipements, etc., la sécurité du personnel et des équipements, le suivi des travaux sous-traités, la gestion des approvisionnements des fournitures et des pièces de rechange, la gestion de la disponibilité des moyens de manutention spéciaux et les différents niveaux d'urgence des travaux de maintenance corrective. Les objectifs de l'ordonnancement de la maintenance peuvent être décomposés de manière hiérarchique en objectifs internes et objectifs externes Figure (2. 10), [Coudert, 2000].



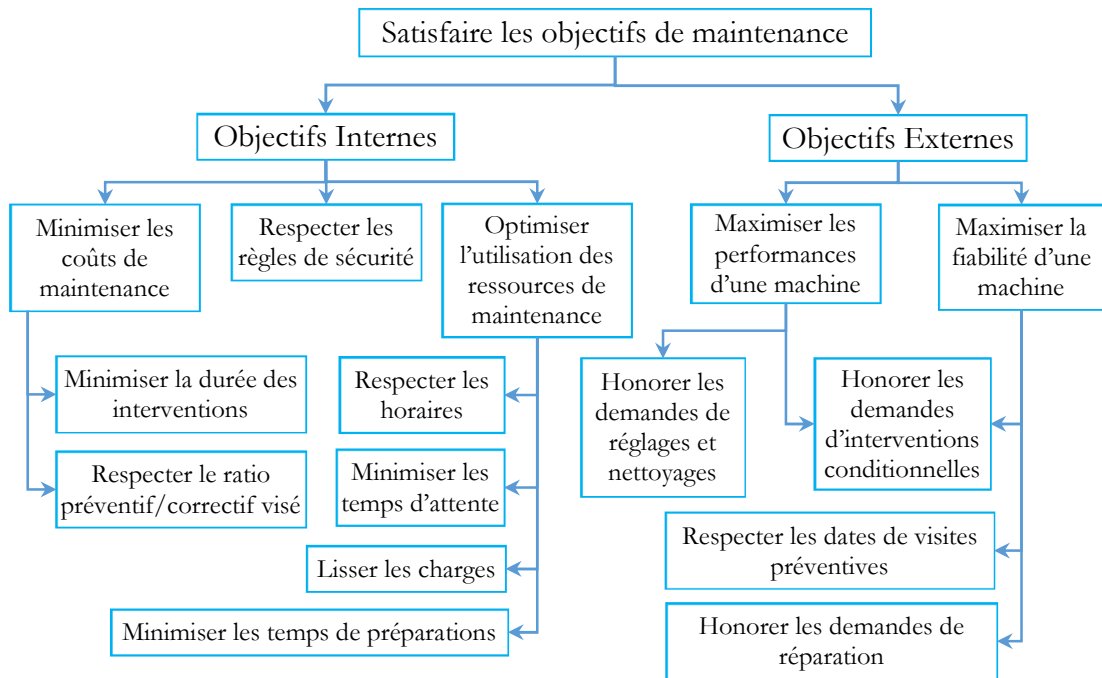


Figure (2.10) : Structure hiérarchique des objectifs de l'ordonnancement de la maintenance

Les objectifs internes concernent le fonctionnement propre du service de maintenance et les objectifs externes sont imposés par d'autres services, principalement par la production. Ils peuvent se retrouver en conflit de par leur nature. En effet, leur satisfaction nécessite des interventions à des dates bien étudiées et qui peuvent être loin des dates optimales.

### 6.5. La fonction réalisation des interventions

La fonction réalisation consiste à accomplir les interventions préventives ou correctives nécessaires. Ses objectifs sont, soit de s'assurer que le matériel est dans l'état de fonctionnement normal spécifié, soit (ou/et) de l'y remettre s'il n'y est pas, cela en respectant les délais prescrits et en utilisant les moyens mis à disposition. Elle commence à la date fixée par l'ordonnancement et utilise les procédures rédigées par la préparation.

La réalisation peut être effectuée par du personnel interne à l'entreprise ou partiellement ou totalement sous-traitée. Le responsable de la réalisation aura pour tâche de coordonner les travaux, de veiller au respect des consignes et de contrôler la conformité et la qualité des interventions.

## 7. Coût de la maintenance

La maintenance souvent qualifiée d'un poste générateur de dépenses. C'est juste, mais peut être en même temps un vrai gisement de productivité, ce qui devient actuellement de plus en plus un fait établi, mais moins visible. La maintenance est donc un champ d'action privilégié de recherche

d'amélioration des coûts de production, et c'est la seule source significative de compétitivité intrinsèque, car, elle ne repose que sur les compétences de l'entreprise à la maîtrise de l'ensemble de son système de production.

Les coûts de maintenance représentent l'ensemble des dépenses engagées pour maintenir un système, en état de fonctionnement. Ils sont Souvent liés aux exigences de sécurité, de fiabilité ou encore de disponibilité. En effet, l'optimisation et l'analyse des couts de maintenance sont devenus décisifs quant à la détermination du prix de revient des produits.

L'optimisation de la maintenance consiste à trouver un équilibre optimal entre les coûts issus de la maintenance préventive et les coûts générés par la maintenance corrective tout en respectant les objectifs fixés. L'entreprise doit rechercher, alors, un compromis afin d'optimiser les relations entre les coûts de maintenance liés à l'investissement humain et matériel, et les pertes consécutives aux arrêts de la production. La Figure (2.11) illustre les effets de la fréquence des opérations de maintenance sur les coûts liés soit à la maintenance corrective, soit à la maintenance préventive.

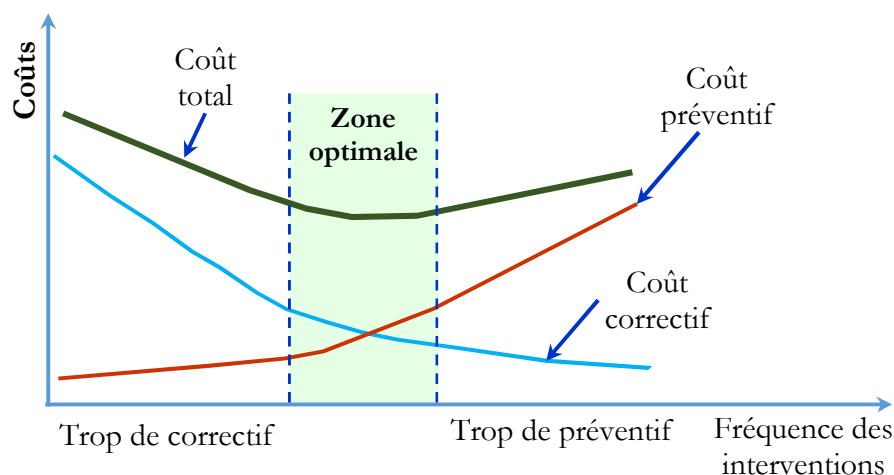


Figure (2.11) : équilibre maintenance corrective/ préventive

L'analyse des coûts de maintenance permet au responsable du service maintenance d'effectuer des choix cruciaux, concernant l'établissement d'un budget prévisionnel annuel, le suivi des dépenses en accord avec le respect du budget, le niveau de maintenance préventive à mettre en œuvre, la vérification de l'efficacité des actions de maintenance, le recours ou non à sous-traitance d'une intervention et à la main d'œuvre externe. L'analyse permet aussi de choisir le type de renouvellement de l'équipement, soit par remplacement à l'identique ou non, soit remise à niveau ou par reconstruction.

## 7.1. Coûts de la maintenance corrective

Les principales dépenses relatives à la maintenance peuvent être associées à deux catégories de coûts ; coûts directs et coûts indirects.

### 7.1.1. Les coûts directs ( $C_{Dm}$ )

- Coûts de main d'œuvre  $C_{mD}$ ,
- Frais généraux du service maintenance,
- Coûts de possession des stocks, des outillages et des machines,
- Consommation de matières, de fournitures, de produits utilisés,
- Coût de consommation des pièces de rechange,
- Coûts des contrats de maintenance,
- Coûts des sous-traitances

Les coûts directs de maintenance ( $C_{Dm}$ ) peuvent être regroupés en quatre classes : les dépenses de main d'œuvre ( $C_{mD}$ ), les dépenses fixes du service maintenance ( $C_{fD}$ ), les dépenses de consommables ( $C_{cD}$ ) et les dépenses externalisées ( $C_{eD}$ ).

$$C_{Dm} = C_{pD} + C_{fD} + C_{cD} + C_{eD} \quad (2.4)$$

### 7.1.2. Les coûts indirects

Les coûts indirects de maintenance ( $C_{Im}$ ) peuvent aussi être nommés coûts indirects de disponibilité ou coûts indirects d'arrêt de production. Ces coûts intègrent toutes les conséquences économiques induites par l'arrêt d'un équipement requis [Efaga, 2004], [Souris, 1990].

- Perte de production ( $C_{pI}$ ),
- Coûts de la main d'œuvre de production ( $C_{mI}$ ).
- Coûts des arrêts induits ( $C_{aI}$ ),
- Coûts de la non-qualité, des rebuts et des délais non tenus ( $C_{qI}$ ),
- Frais de redémarrage de production ( $C_{rI}$ ),
- Coûts induits en cas d'accident corporel ( $C_{iI}$ ) :

$$C_{Im} = C_{pI} + C_{mI} + C_{aI} + C_{qI} + C_{rI} + C_{iI} \quad (2.5)$$

Coût de maintenance corrective ou coûts de défaillance ( $C_{mc}$ ), résultent des coûts directs et indirects d'une ou d'un cumul de défaillances relatives à un équipement, [Diebolt, 2007].

$$C_{mc} = C_{Dm} + C_{Im} \quad (2.6)$$

## 7.2. Coûts de la maintenance préventive

Les coûts de la maintenance préventive ( $C_{mp}$ ) varient logiquement à l'inverse des coûts directs de maintenance corrective ( $C_{mc}$ ). En fait, la stratégie de maintenance permet de choisir librement le niveau de préventif à organiser sur un équipement. Dans le cas d'un niveau élevé de préventif, les coûts correctifs deviennent résiduels.

### 7.2.1. Coût de la maintenance préventive systématique :

Le coût de la maintenance préventive d'un équipement (sur une période donnée) peut s'exprimer par l'équation suivante :

Coût total = Coût du préventif systématique + Coût des défaillances résiduelles

$$C_{total} = \frac{\tau}{\tau^*} C_{mp} + \lambda \times \tau \times C_d \quad (2.7)$$

- $\tau$  : Période de référence exprimée en heures
- $\tau^*$  : Périodicité d'intervention systématique
- $\frac{\tau}{\tau^*}$  : Nombre d'interventions préventives systématiques pendant la période de référence.
- $\lambda$  : Taux de défaillance résiduelle, en défaillances par heure
- $C_{mp}$  : Coût d'une intervention préventive systématique.
- $C_d = C_{Dm} + C_{ind}$  : Coût d'une défaillance résiduelle avec  $C_{int} = t_h \times T_{int}$
- $C_{int}$  : Coûts d'indisponibilité.
- $t_h$  Le taux horaire (unité monétaire /heure)
- $T_{int}$  : Temps d'intervention corrective.

Les coûts de maintenance préventive systématique et Les coûts de maintenance corrective seront utilisés dans le dernier chapitre pour formuler la fonction objectif concernant la maintenance.

## 8. Les attributs de la sûreté de fonctionnement.

La Sûreté de Fonctionnement est l'aptitude d'un système à satisfaire à une ou plusieurs fonctions requises dans des conditions données. En effet, pour s'assurer qu'un système manufacturier de production garanti cette aptitude, il est capital de maîtriser les machines qui le compose. Les équipements composant ces machines sont caractérisés par des indicateurs fondamentaux de sûreté de fonctionnement. Ces indicateurs liés aux notions de fiabilité, maintenabilité et disponibilité (FMD), permettent d'aider la prise de décision sur le choix de politique de maintenance à appliquer et s'attachent à suivre le bon déroulement des opérations de maintenance. Ainsi, pour une maintenance

préventive, deux indicateurs sont principalement suivis : le temps d'intervention et la périodicité. Pour une maintenance corrective, le temps de réparation et la fréquence des défaillances sont les paramètres prépondérants en termes de sûreté de fonctionnement.

### 8.1. Fiabilité

Par définition, la fiabilité  $R(t)$  d'un système (R vient de l'anglais Reliability) est la probabilité de bon fonctionnement du système sur la durée  $[0, t]$  en supposant qu'il n'est pas défaillant à l'instant  $t=0$ . Les indicateurs liés à la fiabilité sont :

$\lambda(t)$  : Taux de défaillance qui représente le nombre de défaillances par unité de temps.

MTBF : temps moyen entre deux défaillances (Mean Time Between Failures).

MTTF : temps moyen avant la première défaillance (Mean Time To failure). pour un système non réparable, on a  $MTTF=MTBF$ .

### 8.2. Disponibilité

Selon la NF EN 13306 la disponibilité : « L'aptitude d'une entité 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 des moyens extérieurs nécessaires de maintenance soient assurés ».

La disponibilité (availability)  $A(t)$  est définie comme étant la probabilité de bon fonctionnement d'un système à l'instant  $(t)$ . Comme indicateurs de disponibilité, on a :

DO : indicateur de base de la disponibilité opérationnelle, qui est le rapport entre le temps moyen de bon fonctionnement sur le temps requis. DO s'exprime par :

$$DO = \frac{MTBF}{MTBF + MTTR} < 1 \quad (2.8)$$

### 8.3. Maintenabilité

Selon l'AFNOR X-06-501, c'est l'aptitude d'un dispositif à être maintenu ou rétabli dans un état dans l'état dans lequel il puisse accomplir une fonction requise lorsque la maintenance est accomplie dans des conditions d'utilisation données avec des moyens et procédures prescrits.

La maintenabilité  $M(t)$  est la probabilité de rétablir le fonctionnement du système après sa défaillance, dans des conditions spécifiques et dans un temps  $(t)$  donné et limité. Les indicateurs de maintenabilité sont :

$\mu(t)$  : taux de réparation qui s'assimile à la probabilité que le système soit réparé entre l'instant  $t$  et l'instant  $(t + dt)$ , sachant qu'il n'est pas réparé sur l'intervalle  $[0, t]$ .

MTTR : moyenne des de réparations (Mean Time To repair).

#### 8.4. Les temps estimateur moyens.

Les définitions ci-après précisent le sens de sigles anglo-saxons, souvent confondus, qui à part les deux premiers, ne s'appliquent qu'aux équipements réparables.

– **Mean Time To Failure** : MTTF (moyenne des temps jusqu'à défaillance)

Ce sigle, quelle que soit l'entité, réparable ou non, doit être interprété comme la moyenne des temps jusqu'à la défaillance totale (si l'entité est redondée) ou jusqu'à la première défaillance sinon.

– **Mean Time Between Failure** : MTBF (moyenne des temps entre défaillances)

Ce sigle est correctement traduit par moyenne des temps entre défaillances (consécutives, dans le cas d'une entité réparée) et non par moyenne des temps de bon fonctionnement.

– **Mean Time To Repair** : MTTR (moyenne des temps techniques de réparation)

Les Anglais utilisent aussi le sigle MART (Mean Active RepairTime).

– **Mean Up Time** : MUT (temps moyen de disponibilité, après réparation)

Ce temps moyen est un peu inférieur au MTTF, car ici la réparation peut être partielle, notamment dans le cas de structures redondantes, alors que le MTTF caractérise la durée moyenne de fonctionnement d'une entité complètement restaurée (comme à  $t = 0$ ).

– **Mean Down Time** : MDT (temps moyen d'indisponibilité, après réparation)

Ce temps inclut : les temps techniques de détection, localisation et réparation de la panne, mais aussi les délais administratifs de gestion et de transport. Le MDT est donc supérieur au MTTR, mais en général faible devant le MUT, c'est pourquoi la différence entre

MTTF et MTBF n'est pas très significative.

– **Rapports entre estimateurs**

La Figure (2.12) illustre les rapports entre estimateurs. Notons que pour une entité non redondée, à  $\lambda$  constant :  $MTBF = MUT + MDT = 1/\lambda$

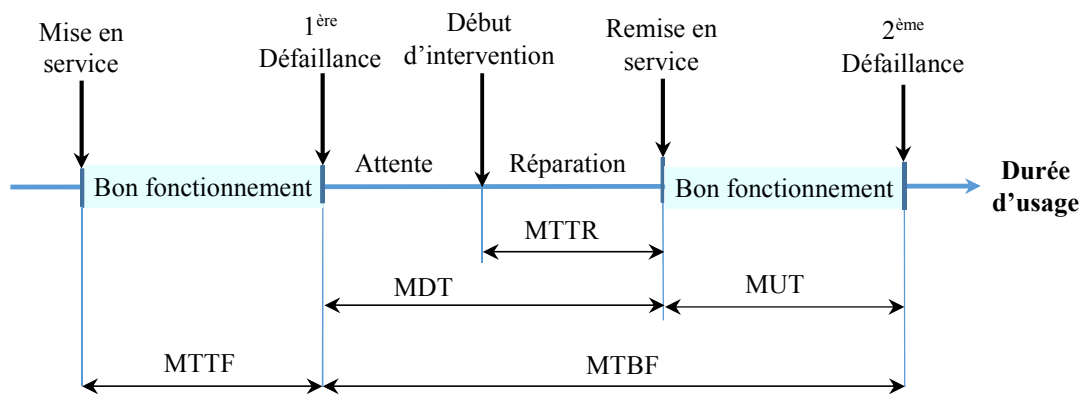


Figure (2.12) : Durées caractéristiques des indicateurs de Fiabilité – Maintenabilité - Disponibilité

### 9. Lois continues usuelles utilisées en fiabilité.

Les défaillances sont à la maintenance ce que les pathologies humaines sont à la médecine : leur raison d'exister. Or, toute défaillance est le résultat d'un mécanisme pathologique rationnel et explicable, dû à une ou plusieurs causes à identifier.

La connaissance du comportement du matériel en exploitation doit permettre de trouver des solutions d'amélioration, ceci par la connaissance des types et des modes de défaillances retrouvés, par analyse méthodique, l'origine de la défaillance.

Les passages d'un état de fonctionnement normal à un état défaillant pouvant se manifester en fonction du temps de manière progressive, soudaine ou de façon Aléatoire, la fiabilité ne connaît pas la notion de défaillance partielle ou progressive, Zwingelstein (1995).

La Figure (2.13) représente trois cas conduisant tous à une défaillance.

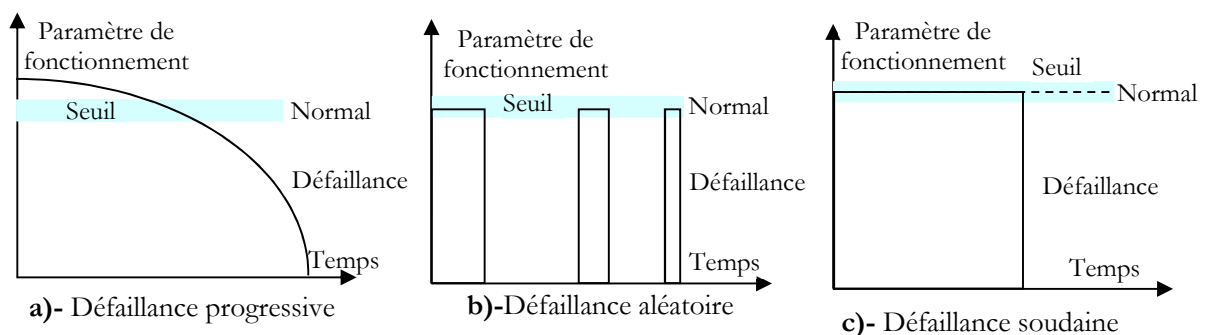


Figure (2.13) : Passage d'un état de bon fonctionnement à l'état de panne

## 9.1. Loi exponentielle

### 9.1.1. Domaine d'utilisation.

La courbe en baignoire de la Figure (2.14), relative à la durée de vie d'un système, fait apparaître une période de maturité plus ou moins longue pendant laquelle le taux de défaillance d'un système ou d'un composant est sensiblement constant :

C'est le champ d'application de la loi exponentielle, qui repose sur l'hypothèse  $\lambda = \text{constante}$ . Dans ce cas, les défaillances aléatoires émergent sous l'action de causes diverses et indépendantes par opposition à la période suivante, dite de vieillesse, pendant laquelle un ou plusieurs modes de défaillance prédominants apparaissent (fatigue, corrosion, etc.). Le domaine de l'électronique se prête bien à l'utilisation de la loi exponentielle, dès lors que les composants sont déverminés.

La plupart des analyses prévisionnelles de fiabilité et de maintenabilité reposent sur les hypothèses exponentielles :

$\lambda$  taux de défaillance constant en fiabilité, donc  $MTBF = 1/\lambda$ ;

$\mu$  taux de réparation constant en maintenabilité, donc  $MTTR = 1/\mu$

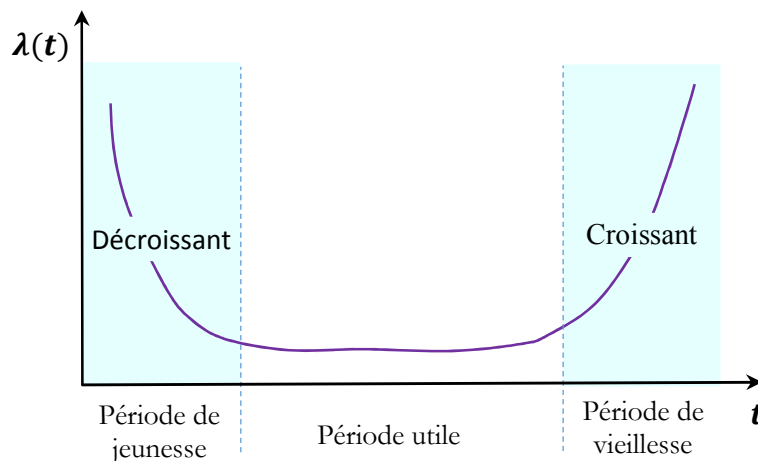


Figure (2.14) : Courbe en baignoire, Rémi (2011)

Cette courbe fait apparaître trois phases :

**Zone A** : Phase de jeunesse : période de défaillance précoce ou jeunesse. Le taux de défaillance diminue (élimination des défauts liés à la fabrication).

**Zone B** : Phase de maturité, période de défaillances aléatoires indépendantes à taux constant. Les défaillances sont liées à une usure normale.



**Zone C** : Phase d'obsolescence : défaillances d'usure ou pannes de vieillesse. Le taux de défaillance augmente. De nombreuses pièces ont atteint une usure anormale.

### 9.1.2. Caractéristiques de Loi exponentielle

Elle se retrouve à partir de la loi fondamentale de la fiabilité avec  $\lambda$  constant.

La probabilité de survie entre 0 et  $t$  devient :  $R(t) = e^{-\lambda t}$  Figure (2.15-a)

- Densité de probabilité :  $f(t) = \lambda e^{-\lambda t}$  . Figure (2.15-b)
- Taux instantané de défaillance  $\lambda(t) = \frac{f(t)}{R(t)} = \lambda_0$  , Figure (2.15-c)

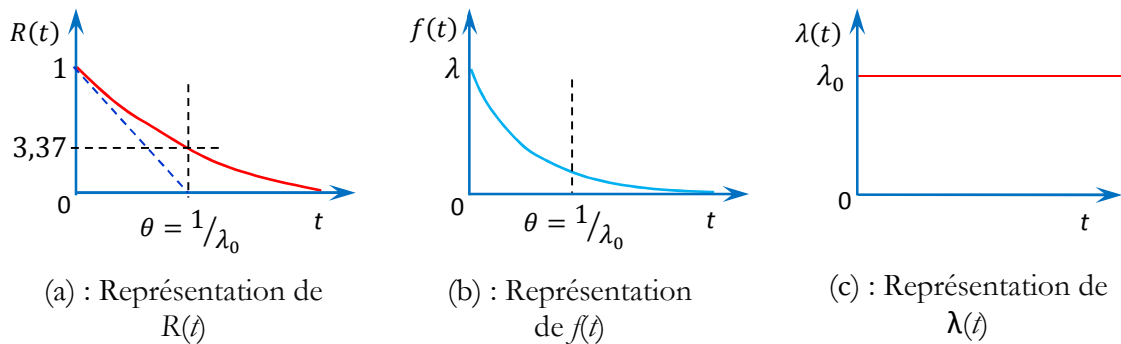


Figure (2.15) : Représentation graphique de la loi exponentielle

- Fonction de répartition :  $F(t) = 1 - R(t) = 1 - e^{-\lambda t}$  = probabilité de panne entre 0 et  $t$ .
- Espérance mathématique :  $E(t) = 1/\lambda = MTBF$ .
- Écart-type :  $\sigma = 1/\lambda$
- Durée de vie associée à un seuil de fiabilité : tirons  $t$  de la loi exponentielle :

$$-\lambda t = \ln(R(t)) \quad \text{soit} \quad t = \frac{1}{\lambda} \ln\left(\frac{1}{R(t)}\right) \quad (2.9)$$

## 9.2. Loi de Weibull

### 9.2.1. Domaine d'utilisation de la loi de Weibull, [Frédéric, 2015]

La loi de Weibull à trois paramètres est très « souple », ce qui lui permet de s'ajuster à un grand nombre d'échantillons prélevés au long de la vie d'un équipement. Elle couvre les cas de taux de défaillance variables, décroissants (période de jeunesse) ou croissants (période de vieillesse). Son exploitation fournit :

- Une estimation de la *MTBF* de la population de défaillances ;

- les équations de  $R(t)$  et de  $\lambda(t)$  ainsi que leurs variations sous forme graphique ;
- Mais aussi le paramètre de forme  $\beta$  qui peut orienter un diagnostic, sa valeur étant caractéristique de certains modes de défaillance.

$\beta$  : est appelé paramètre de forme ( $\beta > 0$ ) : suivant que  $\beta$  est supérieur, égal ou inférieur à 1, la loi de Weibull correspond à un taux de défaillance instantané croissant, constant ou décroissant.

$\eta$  : est appelé paramètre d'échelle ( $\eta > 0$ ) : parfois nommé caractéristique de vie, c'est un simple paramètre de changement d'échelle de temps.

$\gamma$  : est appelé paramètre de position ( $-\infty < \gamma < +\infty$ ) : il définit un changement d'origine dans l'échelle des temps.

### 9.2.2. Expressions mathématiques caractéristiques de la loi de Weibull, [Frédéric,2015]

- Densité de probabilité  $f(t)$ . , Figure (2.16-b)

$$f(t) = \frac{\beta(t - \gamma)^{\beta-1}}{\eta^\beta} \times \exp \left[ - \left( \frac{(t - \gamma)}{\eta} \right)^\beta \right] \quad (2.10)$$

On note :

- $t_0$  (ou  $\gamma$ ) le paramètre de centrage de l'origine du temps ;
- $\beta$  un paramètre  $> 0$  de forme ;
- $\eta$  un paramètre d'échelle, homogène à un temps.

C'est la valeur de  $\lambda$  qui fixe la période du taux de défaillance couverte par la loi, [Zwingelstein, 1995] :

- si ( $\beta < 1$ ), le taux instantané est décroissant, les défaillances sont précoces et le système s'améliore ;
- si ( $\beta = 1$ ), le taux est constant, la loi se ramène à la loi exponentielle comme cas particulier de la loi de Weibull ;
- si ( $\beta > 1$ ), le taux est croissant, on est en période d'usure.

- Fonction de répartition  $F(t)$  et fiabilité  $R(t)$  : Figure (2.16-a)

$$F(t) = 1 - \exp \left[ - \left( \frac{(t - \gamma)}{\eta} \right)^\beta \right] \quad (2.11)$$

La fiabilité correspondante est donc  $R(t) = 1 - F(t)$

$$R(t) = \exp \left[ - \left( \frac{(t - \gamma)}{\eta} \right)^\beta \right] \quad (2.12)$$

- Expression du taux instantané de défaillance  $\lambda(t)$  : Figure (2.16-c)

$$\lambda(t) = \frac{f(t)}{1 - F(t)} \quad (2.13)$$

$$\lambda(t) = \frac{\beta}{\eta} \left( \frac{1 - \eta}{\gamma} \right)^{\beta-1} \quad \text{avec} \quad \begin{cases} t \geq \gamma \\ \beta > 0 \\ \eta > 0 \end{cases}$$

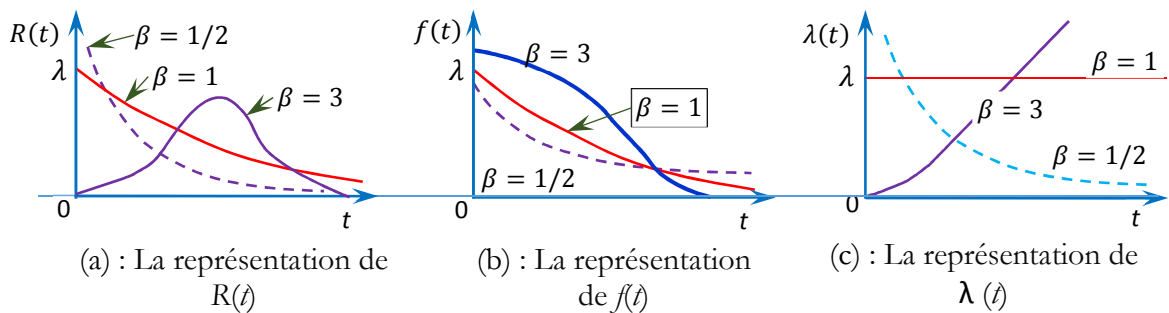


Figure (2.16) : La représentation graphique de la loi Weibull.

- La valeur moyenne et la variance sont données par :

$$\begin{cases} m = \gamma + \eta \Gamma \left( \frac{1 + \beta}{\beta} \right) \\ \text{var}(t) = \eta^2 \left[ \Gamma \left( \frac{2}{\beta} + 1 \right) - \Gamma^2 \left( \frac{1 + \beta}{\beta} \right) \right] \end{cases} \quad (2.14)$$

$$\text{avec} \quad \Gamma(b) = \int_0^{+\infty} x^{b-1} \exp(-x) dx \quad (2.15)$$

- Justification mathématique de la structure du papier de Weibull :

$$F(t) = 1 - \exp \left[ - \left( \frac{(t - \gamma)}{\eta} \right)^\beta \right]$$

$$1 - F(t) = \exp \left[ - \left( \frac{(t - \gamma)}{\eta} \right)^\beta \right] \quad (2.16)$$

Nous pouvons prendre le log népérien des deux membres car  $R(t) \leq 1$ .

$$\frac{1}{R(t)} = \frac{1}{1 - F(t)} \geq 1$$

$$\ln\left(\frac{1}{1-F(t)}\right) \geq 0$$

L'équation devient :

$$\ln\left(\frac{1}{1-F(t)}\right) = \left(\frac{t-\gamma}{\eta}\right)^\beta$$

$$\ln\left(\frac{1}{1-F(t)}\right) = \beta \times \ln\left(\frac{t-\gamma}{\eta}\right)$$

$$\ln\left(\frac{1}{1-F(t)}\right) = \beta \times \ln(t-\gamma) - \beta \times \ln(\eta) \quad (2.18)$$

Qui est de la forme :

$$Y = \beta \cdot X + C \quad (2.19)$$

Nous obtenons une relation linéaire entre  $X$  et  $Y$  telle que la droite  $D_1$  de régression des nuages de points  $(F_i, t_i)$  a pour pente le paramètre  $\beta$ . Au point  $t = \eta = 1$ , origine du repère  $(X, Y)$ ,  $\ln \eta = 0$ , donc  $C = 0$ . La droite  $D_2$  parallèle à  $D_1$  passant par l'origine de  $(X, Y)$  a pour équation  $Y = \beta \times X$ . Elle permet de déterminer la valeur de  $\beta$  sur l'axe b.

## 10. Les stratégies de l'ordonnancement de la production et de la maintenance.

Trois stratégies d'ordonnancement de la production et de la maintenance ont été recensées dans la littérature : stratégie d'ordonnancement séparé, séquentiel et intégré, [lee et al., 2000]. Ces stratégies visent à résoudre, le plus efficacement possible, les conflits structurels, qui existent entre les activités de la fonction production et la fonction maintenance :

### 10.1. Stratégie de l'ordonnancement séparé

Actuellement, la maintenance et la production sont le plus souvent traitées de manière indépendante au sein de l'entreprise [Bembla, 2002]. Les ordonnancements correspondants à ces deux activités sont donc réalisés de manière séparée et interfèrent bien souvent l'un avec l'autre entraînant des retards dans la production ou dans la maintenance. Cette méthode implique la mise en place d'une communication accrue entre les services de maintenance et de production pour limiter les conflits dans l'immobilisation des ressources aussi bien humaines que matérielles, Figure (2.17).

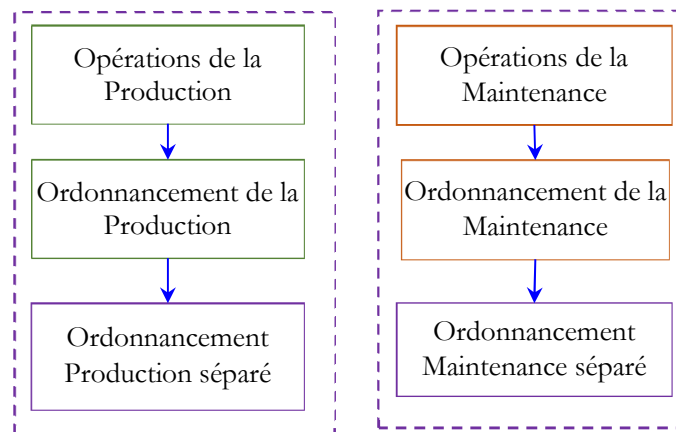


Figure (2.17) : Ordonnancement Séparé

## 10.2. Stratégie de l'ordonnancement séquentiel

Cette politique consiste à planifier l'une des deux activités, maintenance ou production, et à utiliser cet ordonnancement comme une contrainte supplémentaire d'indisponibilité des ressources dans la résolution du problème d'ordonnancement de l'ensemble des deux types de tâches Figure. (2.18). De manière générale, la maintenance est planifiée en premier, par la suite l'ordonnancement de la production est réalisé en prenant les opérations de maintenance comme des contraintes fortes d'indisponibilité des ressources [Aggoune, 2002].

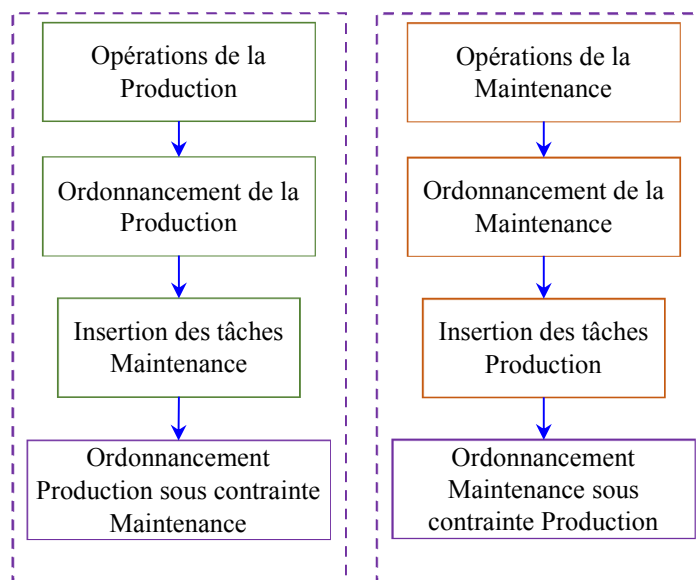


Figure (2.18) : Ordonnancement Séquentiel

### **10.3. Stratégie de l'ordonnancement intégré**

#### **10.3.1. Définition de l'intégration**

Naturellement, plusieurs définitions de l'intégration existent. Nous avons proposé la définition suivante : « intégrer c'est établir et renforcer les liens qui peuvent exister entre deux ou plusieurs entités de base afin d'en déduire une et une seule entité qui agrège les entités précédentes et préserve dans son comportement la cohérence intra et inter entités » [Talbi, 1997]. Cette définition peut être complétée, en considérant l'intégration comme "une démarche globale de simplification et d'organisation du système de production dans un objectif de décloisonnement des fonctions de l'entreprise et de synchronisation des tâches qui concourent à la réalisation des produits. Elle doit s'inscrire dans une vision globale du plan de développement stratégique de l'entreprise qui prend en compte, ses priorités, ses objectifs, ses contraintes, son existant et l'évolution de son environnement" [Vernadat, 1996].

#### **10.3.2. Politique d'intégration.**

Cette politique consiste à créer un ordonnancement conjoint et simultané (intégré) des tâches de maintenance et de production [Brandolese, 1996] Figure (2.19). Une telle politique de planification limite les risques d'interférence entre la production et la maintenance et permet ainsi d'optimiser la qualité des ordonnancements. Vu la différence de caractérisation des tâches de maintenance et de production. Néanmoins, elle offre un bon espoir de voir un jour disparaître les conflits d'utilisation des ressources, mais elle implique la fusion des deux services production et maintenance, ou au moins la création d'un lien très fort entre les deux. De plus, un nombre non négligeable de problèmes est posé par la différence de caractérisation entre les tâches de maintenance et de production. Les premières possèdent bien souvent une date de début au plus tard en fonction des risques admissibles et ne possèdent pas de durée opératoire connue avec précision mais juste une durée moyenne (MTTR). Les secondes, quant à elles, sont caractérisées par une date de début au plus tôt, une date échue et une durée opératoire déterminée. Voilà autant d'obstacles à une planification commune aisée de ces deux types d'activités.

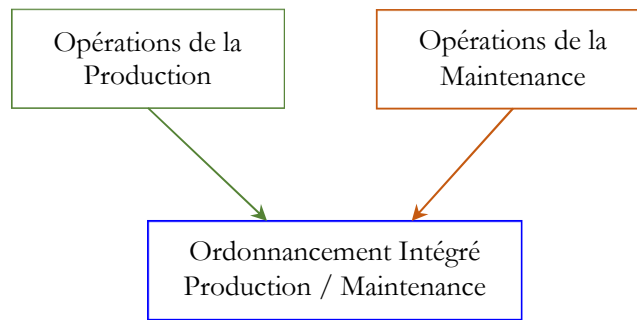


Figure (2.19) : Ordonnancement Intégré

Ordonnancer simultanément les tâches production et maintenance préventives permettra certainement d'améliorer la productivité de l'entreprise et d'éliminer les conflits inhérents à la nature et aux objectifs de ces deux fonctions.

## 11. Intégration des fonctions production et maintenance

La nécessité de faire partager les ressources entre les fonctions production et maintenance est, généralement, une source de conflits et traduit l'existence d'un lien fort entre ces fonctions. Ce lien se renforce et se concrétise à chaque niveau du processus de préparation à la fabrication. En effet, trop souvent, les séquences d'opérations de maintenance et de production sont perçues comme antagonistes. Les périodes d'immobilisation des équipements nécessaires aux interventions des agents de maintenance, sont considérées comme des perturbations réduisant les périodes d'utilisation des ressources et non comme un facteur favorisant leur bon déroulement. Ce type de conflit entraîne naturellement des querelles qui nuisent à la productivité globale de l'entreprise. Ainsi, l'intégration de ces deux fonctions s'impose, et vise les objectifs suivants :

- simplifier les flux physiques et informationnels entre les services maintenance et production,
- augmenter la disponibilité des équipements (planification et gestion des interventions),
- maîtriser les procédés (réduction des dysfonctionnements, ...),
- mieux gérer les ressources en planifiant l'ensemble des activités liées aux processus de l'entreprise,
- mieux orienter la stratégie d'achat (choix des fournisseurs et des méthodes d'approvisionnement),
- diminuer le niveau des stocks (méthodes de gestion des stocks, gestion à flux tendu),
- simplifier et accélérer les flux physiques et informationnels liés aux ressources et produits,
- décloisonner l'organisation (définir et coordonner tout type d'échange entre services),
- justifier et orienter au mieux les investissements futurs (définir les besoins et les priorités, élaborer une stratégie d'achat, ...),

L'intégration des fonctions production et maintenance est basée essentiellement sur l'élaboration d'un planning global qui intègre le planning production, destiné à satisfaire les besoins client (coût, délai, qualité, quantité, ...) et le planning maintenance ayant pour but d'optimiser la disponibilité des ressources en réduisant, sous des contraintes de coûts les arrêts de production.

La réalisation du planning global est menée de la façon suivante :

- définition des tâches de maintenance et de production à intégrer selon les différents horizons temporels,
- gestion des tâches prioritaires et allocation des ressources,
- identification des règles de décision et leurs périmètres de validité pour la planification des tâches production et maintenance,
- prise en compte des contraintes et des aléas de production et de maintenance,
- évaluation du niveau de performance du nouveau planning.

Afin d'intégrer les différents aléas du système de production, un réajustement du planning doit être réalisé lors de l'élaboration du programme d'ordonnancement. La méthode de planification intégrée a pour but non seulement d'éviter sur un même équipement le chevauchement des tâches production et maintenance, mais d'optimiser aussi leur synchronisation par l'insertion des tâches maintenance dans le planning production.

### **11.1. La situation conflictuelle des fonctions production et maintenance**

Dans la littérature concernant l'ordonnancement de la production, les ressources (machines) sont considérées dans la plupart des cas comme disponibles à tout moment ou éventuellement durant certaines fenêtres de temps. Dès lors, le service maintenance d'une entreprise n'est jamais prioritaire sur la production, la demande de l'immobilisation d'une ressource pour effectuer une opération de maintenance préventive, est généralement ressenti par les responsables de la production comme une perturbation supplémentaire et forcément malvenue, en particulier avec un programme de production chargé (régime de travail 3x8 continu) et pressant par les clients. L'arrêt de la ressource est généralement planifié indépendamment du service production et sans considération de ses effets sur la productivité. D'autre part de point de vue maintenance, les interventions de maintenance préventive ont un rôle prépondérant, puisqu'elles permettent au système de production de fonctionner de façon nominale. En effet, le coût engendré par une panne impliquant une maintenance corrective, un arrêt non programmé de la production, des retards conséquents de livraison, etc., est largement supérieur à celui d'un arrêt prévu de la production.



En réalité, la proposition d'un ordonnancement intégré des activités de la production et de la maintenance peut contribuer énormément quant à l'amortissement de l'état conflictuel. En effet il peut satisfaire simultanément les contraintes de coût, qualité et délais et les contraintes de disponibilité. Quatre facteurs de complexité du problème d'intégration sont à considérer :

- la typologie des systèmes de production : atelier à flux de production unique, parallèle ou hybride,
- les tâches de maintenance : une seule tâche de maintenance périodique par machine, des tâches de maintenance pouvant être préemptées par la production, ou un ensemble de tâches de maintenance préventive par machine,
- l'horizon de planification : à long terme, à moyen terme ou à court terme,
- les compétences des ressources humaines : un seul opérateur de maintenance qui intervient, plusieurs opérateurs de maintenance ou des équipes constituées en fonction des compétences individuelles et des besoins du système de production.

Les fonctions production et maintenance agissent sur les mêmes ressources, cependant l'établissement d'ordonnancement intégrant leurs activités respectives s'avère indispensable pour la concrétisation de l'objectif globale de l'entreprise, tout en optimisant simultanément les objectifs des deux fonctions. Ces conflits entre les fonctions maintenance et production génèrent différentes variantes de l'indisponibilité machines.

## 11.2. Typologie de l'indisponibilité

L'introduction des contraintes d'indisponibilité des ressources rend les problèmes classiques plus réalistes. Bien que de plus en plus les études soient dédiées à ce type de problème, leur nombre reste faible en comparaison avec les problèmes sans périodes d'indisponibilité. S'intéresser à la flexibilité est plus réaliste, mais augmente la complexité des problèmes associés. Cette flexibilité peut être liée à au moins un des points suivants :

- La date de début de la période d'indisponibilité machine, introduite par [Aggoune, 2004] : déplacer la période d'indisponibilité dans sa fenêtre de temps (définie par des dates de début au plus tôt et au plus tard) permet la création d'un temps libre sur la machine pour effectuer une opération soit avant la période d'indisponibilité soit après ; ce qui permet à l'opération de finir plus tôt.
- La durée de la période d'indisponibilité machine : en fonction d'une décision de management, une priorité peut être donnée à la production. En effet, il est possible en élaborant un ordonnancement de modifier les durées des périodes d'indisponibilité pour chaque machine

pour qu'elle ait un nombre minimum de périodes d'indisponibilité et qu'elle respecte une durée globale minimum d'indisponibilité.

- L'interruption (préemption) d'une opération par une période d'indisponibilité avec ou sans pénalité : lorsque la préemption est autorisée, une opération de production peut être interrompue par une période d'indisponibilité, ensuite reprise avec une éventuelle pénalité, dès que la machine est à nouveau disponible. Ceci peut être le cas de produits regroupés en lots.
- Une extension du problème étudié, est celui qu'une opération peut être effectuée par un ensemble de ressources, sauf que cette opération ne nécessite qu'une seule ressource à la fois. Dans le problème classique, une opération nécessite exactement une seule ressource définie a priori pour l'effectuer. Lorsque cette ressource est non-disponible, l'opération doit attendre que la ressource redevienne disponible à nouveau. Ce qui n'est pas le cas si plus d'une machine peut exécuter l'opération.

Il existe dans la littérature quatre cas de figure pour lesquels une opération peut être interrompue par une période d'indisponibilité : opérations strictement non-préemptives, sécables (résumable), non-sécables (non-résumable), semi-sécables (semi-résumable). Le premier cas est étudié par [Aggoune, 2004] et les trois autres par [Lee, 1999], [Lee et al. 1997], [Azzem et al. 2011]. Une opération est dite strictement non-préemptive lorsqu'elle ne peut être interrompue ni par une autre opération ni par une période d'indisponibilité. Une opération interrompue par une période d'indisponibilité est dite sécable si son exécution peut continuer aussitôt que la machine qui l'exécute est de nouveau disponible. Elle est dite non-sécable si elle doit recommencer complètement. Il est important de noter que ce cas est différent du cas non-préemptif du fait que dans ce dernier, lorsque l'opération ne peut être effectuée avant la période d'indisponibilité, elle doit commencer et se terminer après. Une opération est dite semi-sécable si elle doit partiellement recommencer lorsque la machine est de nouveau disponible. Noter que l'étude du cas semi-sécable inclue les cas sécables et non-sécable. Cependant, l'étude du cas non-sécable est moins pertinente à considérer que les autres cas ; car nous étudions des périodes d'indisponibilité prévues.

Il existe une autre terminologie introduite dans [Mauguière et al. 2005], Elle concerne les périodes d'indisponibilité permettant l'interruption d'opérations : périodes d'indisponibilité traversable (crossable) et non-traversable (non-crossable). Ainsi, une période d'indisponibilité est dite traversable si elle permet l'interruption d'une opération ; bien entendu, cette interruption ne se fera pas si l'opération est non-préemptive. Une période d'indisponibilité qui ne permet pas l'interruption d'une opération est dite non-traversable ; dans ce cas, l'opération ne sera pas interrompue même si elle est préemptive. [Mauguière et al. 2005] énumèrent tous les cas générés par les caractères sécable

et traversable des opérations et des périodes d'indisponibilité et les classifient pour définir de nouveaux problèmes. Les cas sécables et non-sécable y sont étudiés. L'étude des différentes variantes d'indisponibilité, peut conduire à des solutions à travers un processus de négociation Production / Maintenance.

### 11.3. Le processus de négociation de la production et de la maintenance

Un protocole de négociation efficace nécessite une modélisation commune des degrés de liberté des différentes tâches à planifier sur un horizon temporel donné. Les fenêtres temporelles associées à ces tâches décrivent les degrés de liberté dans leur positionnement temporel. Elles peuvent être attachées aux tâches production et maintenance préventives. Si un conflit de partage de ressources est détecté par le module de planification globale, les agents production et maintenance sont aussitôt informés et un processus de coopération / négociation est lancé Figure (2. 20).

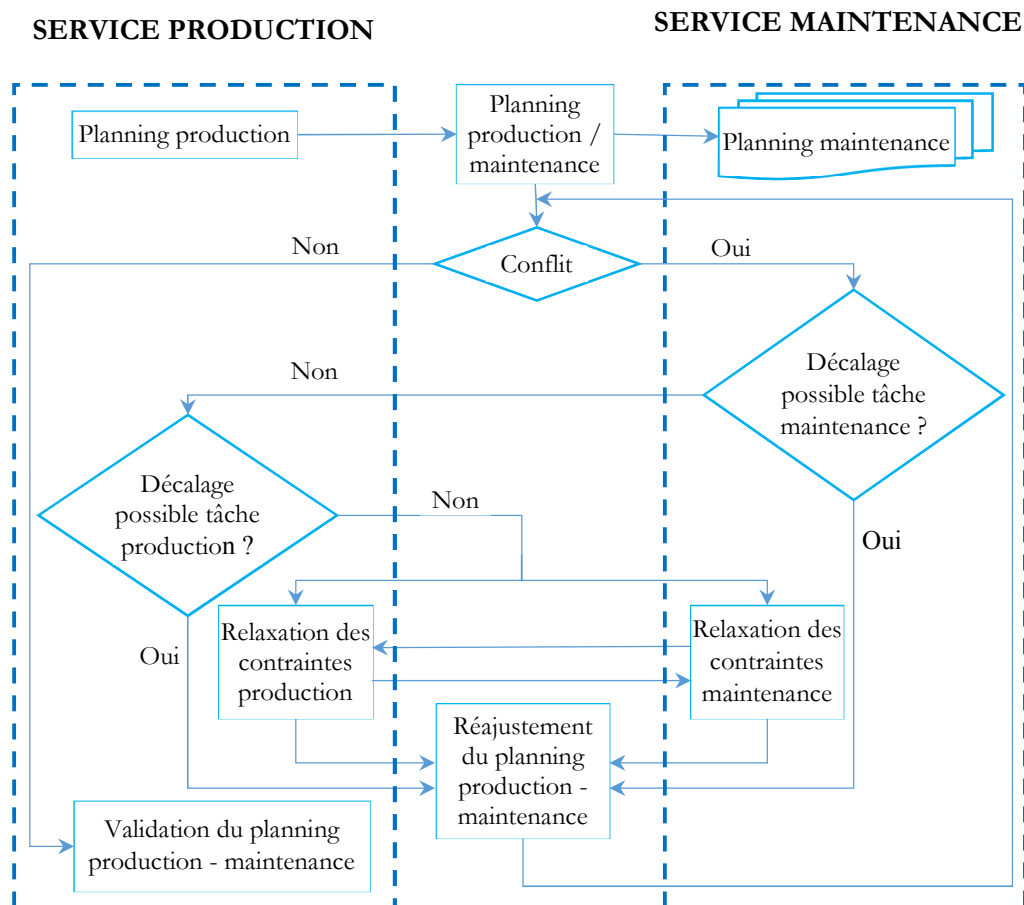


Figure (2. 20) : Processus de négociation production / maintenance

L'agent du service maintenance vérifie s'il est possible de déplacer le début de la tâche maintenance à l'intérieur de sa fenêtre temporelle. Si le début de la tâche reste dans la fenêtre, l'agent maintenance accepte ce changement et la tâche est repositionnée puis validée. Dans le cas contraire, l'agent maintenance demande à l'agent production s'il est possible de déplacer le début de la tâche production à l'intérieur de sa propre fenêtre temporelle. Il suggère une date de fin compatible avec la fenêtre temporelle de la tâche maintenance en conflit ; si cette proposition est possible, la tâche production est repositionnée et le planning est validé. Sinon, il est nécessaire de trouver un compromis en relâchant la contrainte sur l'une des tâches conflictuelles (maintenance ou production) ou sur les deux à la fois. La résolution d'un conflit sur une tâche peut entraîner un ou plusieurs conflits qui se répercutent sur les autres tâches.

Une remise en cause de l'ensemble des tâches nécessite une bonne coordination des agents production et maintenance. À ce niveau, plusieurs solutions peuvent être envisagées. Les scénarios correspondant seront examinés en considérant les indicateurs de performance associés à chaque planning. Un tel processus de négociation s'applique pour toutes les tâches en conflit.

## **12. Intérêts de l'ordonnancement de la production et de la maintenance**

Parmi, les avantages majeurs que présente un ordonnancement de la production et de la maintenance intégré est la réduction des coûts de maintenance et de production. [Alberto, 2012] a réalisé une étude sur un système à machines parallèles dans l'ordonnancement desquelles ils intègrent des interventions de maintenance préventive. Ces dernières sont planifiées d'après la durée écoulée entre deux opérations successives de maintenance. Les coûts de maintenance préventive et ceux induits par les pannes sont à priori connus, les seconds étant bien entendu toujours supposés supérieurs aux premiers. Dans le cas contraire, on ne verrait pas l'intérêt d'introduire une politique de maintenance préventive. Ainsi que l'obtention d'un ordonnancement robuste, qui est susceptible de rester réalisable malgré l'apparition d'imprévus de production. Pour trouver une solution de ce type, on accepte généralement une certaine dégradation du critère d'optimisation afin de conserver un certain degré de liberté qui permet éventuellement des modifications mineures ou l'insertion d'opérations de maintenance, lorsque l'ordonnancement des opérations de production est réalisé, afin d'avoir un ordonnancement robuste. C'est un ordonnancement que l'on peut situer entre un ordonnancement statique, complètement figé, et un ordonnancement dynamique, souvent remis en cause en partie ou totalement. La robustesse d'un ordonnancement conjoint production / maintenance peut être mesurée par sa réactivité à des événements aléatoires tels que des pannes ou encore par la constatation d'une dégradation d'un équipement induisant des interventions de maintenance

corrective ou conditionnelle. Si les indices de fiabilité de la planification de chacune des tâches restent dans des proportions acceptables, on dira alors que l'ordonnancement est robuste [Benbouzid, 2005].

## **Conclusion**

L'objectif de ce chapitre est de situer et de valoriser l'importance de la fonction maintenance dans le système de production. Ainsi, nous retenons que la fonction maintenance est fortement liée à la fonction production. Malheureusement, leurs objectifs sont antagonistes et demandent les mêmes ressources machines. Cette problématique, rencontrée par les deux fonctions, concerne l'affectation de l'ensemble des tâches de production et de maintenance aux ressources machines demande un compromis entre les deux.

Ainsi, nous déduisons que l'ordonnancement des tâches de la maintenance et de la production séparé génère des conflits qui vont à l'encontre de la performance optimale des ressources machines, notamment leurs disponibilités. Pour une meilleure cohérence entre les deux fonctions, et afin d'éviter ainsi les conflits, le concept d'intégration s'avère indispensable.

---

**CHAPITRE 3.**

**PROBLÈMES D'OPTIMISATION  
MULTICRITÈRES**

# Chapitre 3.

## Problèmes d'Optimisation Multicritères

---

**Résumé :** Dans ce chapitre, nous présentons, au début, un ensemble de définitions et de concepts liés aux problèmes d'optimisation multicritères. Ensuite, nous exposons la problématique issue de ces problèmes et les deux types de classification des approches de résolution : approche non-Pareto composées de méthodes scalaire et non-scalaire et approche Pareto. Ensuite, nous présentons un large éventail de ces méthodes en essayant d'apporter des critiques sur chacune d'elles. Pour terminer, nous présentons les notions de base sur la complexité des algorithmes d'optimisation.

---

## Introduction

Les gestionnaires des entreprises se heurtent quotidiennement à des problèmes technologiques de complexité grandissante, qui surgissent dans des secteurs très divers. Le problème à résoudre peut fréquemment être exprimé sous la forme d'une fonction objectif (critère) à optimiser par rapport à un ensemble de paramètres prédéfinis et un ensemble de contraintes à satisfaire. En effet, Il existe deux catégories de méthodes d'optimisation, la première est l'optimisation monocritère, qui se base sur l'optimisation d'un critère unique, dont l'objectif est de trouver la meilleure solution appelée solution optimale qu'est facilement définie suivant une seule performance du problème étudié. La seconde, l'optimisation multicritère, qui optimise simultanément plusieurs critères (objectifs). Ces derniers sont souvent antagonistes. L'objectif dans ce cas est de chercher les meilleures solutions qui satisfassent ces critères. Ces solutions constituent l'ensemble des différents compromis réalisés entre les critères, qui est connu par Pareto-optimal. La résolution d'un problème d'optimisation multicritères, a conduit les chercheurs à proposer des méthodes de résolution, de plus en plus performantes.

Dans ce chapitre, nous présentons les principes de base de l'optimisation mono et multicritères, et aussi, les différentes approches d'optimisation multicritères. D'une part, les approches non-Pareto qui ne traitent pas le problème comme un véritable problème multicritères. Ces approches sont classées en deux catégories, la première contient les méthodes scalaires (de type a priori) transforment le problème multicritère en problème monocritère : les méthodes agrégées, programmation par but, et  $\epsilon$ -contraintes. La seconde contient les méthodes scalaires (de type a posteriori) utilisent des opérateurs qui traitent séparément les différents critères, elles n'utilisent pas la notion de dominance Pareto parmi ces méthodes : sélection parallèle, sélection lexicographique, etc. D'autre part, les approches Pareto (de type a posteriori) utilisent la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée des objectifs contradictoires. Ces approches peuvent être non-élitistes, c'est-à-dire ne conservent pas les individus Pareto optimaux trouvés au cours du temps. Elles maintiennent difficilement la diversité sur la frontière Pareto, d'où la convergence des solutions vers la frontière de Pareto est lente. Elles peuvent être élitistes c'est-à-dire, elles maintiennent une seconde population appelée archive, contenant les solutions non dominées trouvées au cours des différentes générations de l'algorithme évolutionniste. Les individus de cette population participent avec une certaine probabilité à la sélection et à la reproduction de nouveaux individus.

L'efficacité d'un algorithme est généralement liée au temps nécessaire pour trouver les solutions requises. À la fin de ce chapitre nous présentons quelques notions et définitions liées à la complexité des problèmes d'optimisation combinatoire.



## 1. Problèmes d'optimisation

L'optimisation est une activité qui consiste à améliorer les procédés ou les techniques utilisées afin de les rendre plus performant. La résolution de problèmes d'optimisation est devenue un sujet central dans tous les champs d'activité humaine.

Un problème d'optimisation consiste à chercher une instanciation d'un ensemble de variables soumises à des contraintes, de façon à maximiser ou minimiser un critère. Lorsque les domaines de valeurs des variables sont discrets, on parle alors de problèmes d'optimisation combinatoire.

Un problème d'optimisation, noté  $\wp(X, f)$ , est caractérisé par un ensemble réalisable ou admissible  $X$  non-vide et une fonction objectif  $f$  qui associe un scalaire dans  $\mathcal{R}$  à chaque élément  $x$  de l'ensemble  $X$ . Les éléments de l'ensemble  $X$  sont dits solutions réalisables. Résoudre le problème  $\wp(X, f)$  revient à trouver parmi les solutions réalisables, une qui optimise  $f$ , c'est-à-dire dans le cas d'un problème de minimisation, trouver une solution  $x^*$ .

$$\forall x \in X / f(x) \geq f(x^*) \quad (3.1)$$

Une telle solution est dite optimale et sera notée  $x(X, f)$  avec  $X = \{x_1, x_2, \dots, x_n\}$  et  $f = \{f_1, f_2, \dots, f_m\}$ . Les problèmes de maximisation obéissent aux mêmes règles, à quelques changements près. Un problème d'optimisation peut avoir plusieurs fonctions objectifs, il s'agit alors d'un problème d'optimisation multicritères. Ces problèmes seront abordés dans la section 3. L'ensemble réalisable  $X$  est habituellement défini comme partie de  $\mathcal{R}^n$  où  $n$  est un entier positif désignant la taille du problème. Les solutions réalisables peuvent alors être représentées comme des vecteurs dont les  $n$  composantes sont les variables du problème.

L'ensemble  $X$  est délimité couramment par un système d'inégalités appelées contraintes du problème. Les contraintes sont construites à l'aide de combinaisons des variables, et permettent de caractériser les propriétés communes aux solutions de  $X$  afin de les distinguer parmi toutes les solutions de  $\mathcal{R}^n$ . La description de l'ensemble  $X$  est donc implicite.

Les problèmes d'optimisation sont principalement classés suivant le nombre de critères considérés, en problème d'optimisation monocritère ou multicritères. Les sections suivantes sont réservées à l'étude de ces problèmes.

## 2. Problèmes d'optimisation monocritère

Résoudre un problème d'optimisation monocritère consiste à trouver une solution qui maximise ou minimise un critère donné. L'optimum obtenu n'est pas généralement unique, au contraire, il existe un ensemble de solution qui optimise le même critère.

Nous pouvons décrire formellement un problème d'optimisation comme suit : Considérons un problème de minimisation, sachant que le problème de maximisation du critère ( $f$ ) peut être transformé à un problème de minimisation de  $(-f)$ . Soit  $f : \mathcal{R}^n \rightarrow \mathcal{R}$  et  $X$  un ensemble fini de  $\mathcal{R}^n$ , alors, l'ensemble des minimums est :  $X^* = \{\vec{x}^* \in X \mid \forall \vec{x} \in X, f(\vec{x}^*) \leq f(\vec{x})\}$ , et le minimum du problème sera  $f(X^*)$

### 2.1. Variables de décision

Les variables de décision sont des quantités numériques pour les quelles des valeurs sont à choisir. Cet ensemble de  $n$  variables est appelé vecteur de décision :  $(x_1, x_2, \dots, x_n)$  Les différentes valeurs prises par les variables de décision  $x_i$ , constituent l'ensemble des solutions potentielles.

### 2.2. Espace décisionnel et espace objectif

Deux espaces Euclidiens sont considérés en optimisation, le premier c'est l'espace décisionnel, de dimension  $n$ , où  $n$  étant le nombre de variables de décision. Cet espace est constitué par l'ensemble des valeurs pouvant être prise par le vecteur de décision. Le seconde c'est l'espace objectif : l'ensemble de définition de la fonction objectif, généralement défini dans  $\mathcal{R}$ . La valeur dans l'espace objectif d'une solution est appelée coût, ou fitness.

### 2.3. Contraintes

Dans la plupart des problèmes d'optimisation, des restrictions sont imposées par les caractéristiques du problème. Ces restrictions doivent être satisfaites afin de considérer qu'une solution est acceptable. L'ensemble des restrictions est appelé ensemble des contraintes. Il décrit les dépendances entre les variables de décision et les paramètres du problème. On formule usuellement ces contraintes  $c_j$  par un ensemble d'inégalités, ou d'égalités de la forme :  $c_j(x_1, x_2, \dots, x_n) \geq 0$ .

Mathématiquement, un problème d'optimisation s'écrit, équation (3.2) :

$$\begin{aligned} \min_x f(x) \text{ avec } x = \{x_1, x_2, \dots, x_n\} \text{ et } n \geq 1 \\ \text{S. C:} \\ g_i(x) \leq 0, \quad i = \{1, \dots, m\} \\ h_j(x) = 0, \quad j = \{1, \dots, p\} \\ x \in S \text{ et } S \subset \mathcal{R}^n \end{aligned} \tag{3.2}$$

Où  $f$  est la fonction à minimiser, appelée fonction coût, fonction objectif ou critère,  $x$  représente le vecteur des variables d'optimisation,  $g_i$  sont les contraintes d'inégalité et  $h_j$  les contraintes d'égalité et  $S$  l'espace des variables (appelé aussi espace de recherche).  $S$  définit suivant le type des variables considérées : réelles, entières, mixtes (réelles et entières), discrètes, continues, bornées, etc.

La solution  $x_a$  est dite admissible si  $x_a \in S$  et les contraintes d'optimisation  $g_i(x) \leq 0, i = \{1, \dots, m\}$  et  $h_j(x) = 0, j = \{1, \dots, p\}$  sont satisfaites. La solution de (4.1) est l'ensemble des solutions optimales  $\{x^*\}$ .  $x^*$  est un minimum global de  $f$  si et seulement si :  $f(x^*) \leq f(x) \quad \forall x \in S$  et  $x^*$  est un minimum local de  $f$  si et seulement si :  $f(x^*) \leq f(x) \quad \forall x \in S / \|x - x^*\| \leq \varepsilon$  avec  $\varepsilon > 0$

La Figure (3.1) présente un exemple d'une fonction à une variable, avec des minimums locaux et un minimum globaux. Parmi les minimums locaux, celui qui possède la plus petite valeur de  $f$  est le minimum global. Par ailleurs, une fonction multimodale présente plusieurs minimums (locaux et globaux), et une fonction uni-modale n'a qu'un minimum, le minimum global.

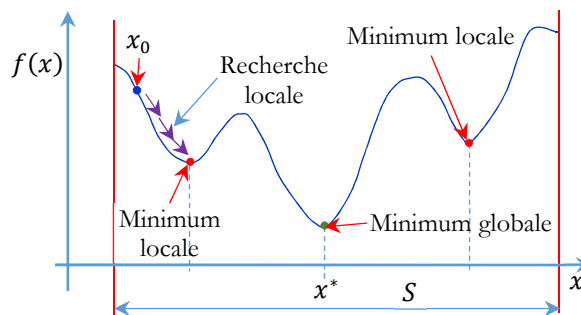


Figure (3.1) : Minimums locaux et minimum globaux d'une fonction à une variable.

## 2.4. Hiérarchisation des méthodes d'optimisation monocritère

Pour résoudre un problème d'optimisation, la littérature nous offre diverses méthodes. Le choix de la méthode adéquate est en fonction de la nature du problème, de sa complexité et de sa taille. Dans la figure (3.2) Nous présentons une hiérarchisation des méthodes d'optimisation monocritère.

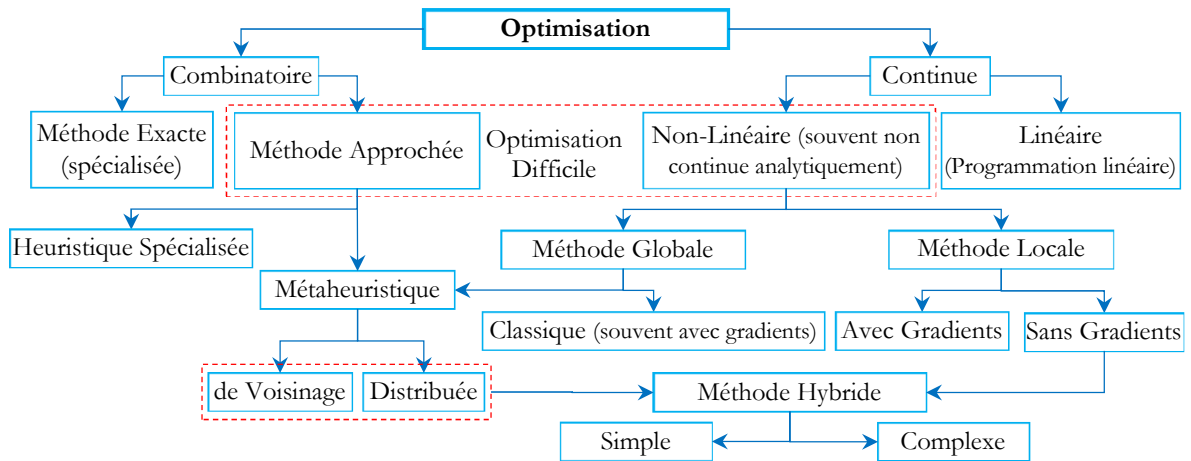


Figure (3. 2) : Classification générale des méthodes d'optimisation monocritère

### 3. Problèmes d'optimisation multicritères

Dans la section précédente, nous avons considéré le cas des problèmes d'optimisation à un objectif unique. Or, dans plupart des cas réels, les problèmes d'optimisation sont décrits à l'aide de plusieurs objectifs souvent contradictoires [Clarisse, 2005], qui devant être optimisés simultanément. Optimiser un tel problème relève donc de l'optimisation combinatoire multicritères.

Bien que, les premiers travaux menés sur les problèmes multicritères furent réalisés au 19<sup>ème</sup> siècle et malgré l'intérêt indéniable de la modélisation et de la résolution multicritères des problèmes rencontrés en industrie, peu de travaux ont été réalisés avant les années 80-90 du 20<sup>ème</sup> siècle. Mais depuis lors, des efforts intensifs ont été déployés pour la résolution de ces problèmes.

L'optimisation multicritères consiste à déterminer, parmi les solutions optimales celles de meilleurs compromis possibles entre les différents critères. Ainsi, ces dernières forment la surface de compromis, dite front de Pareto. Toutes les solutions de cet ensemble sont optimales dans le sens où, aucune solution ne domine les autres pour tous les critères traités. Autrement dit, aucune amélioration ne peut être faite sur l'une des composantes du vecteur-solution sans dégradation d'au moins une autre composante de ce vecteur, [Ulungu, 1993].

#### 3.1. Définition

Un problème d'optimisation combinatoire multicritère (multi-objective combinatorial optimization problem) peut être décrit mathématiquement, comme sur (3.3):

$$\begin{aligned}
 & \text{optimiser } F(x) = (f_1(x), f_2(x), \dots, f_p(x)) \quad p \geq 2 \\
 \text{Sous} \quad & h_j(x) = 0 \quad j = \{1, 2, \dots, m_1\} \\
 & g_k(x) \leq 0 \quad k = \{1, 2, \dots, m_2\} \\
 & x \in \Omega
 \end{aligned} \tag{3.3}$$

Il s'agit d'optimiser simultanément  $p$  fonctions objectifs  $f_i(x)$ , qui représentent l'espace objectif,  $x$  étant un vecteur solution  $\{x_1, x_2, \dots, x_n\}$  d'un espace ( $\Omega$ ) de dimension  $n$ , qui représente les instances de variables de décision  $x_i$ , Figure (3.2). L'espace décisionnel ( $\Omega$ ) représente l'ensemble des solutions réalisables associé à un ensemble de  $m$  contraintes d'égalité  $h_j(x)$  et d'inégalité  $g_k(x)$ . Dans un problème d'optimisation multicritères, il y a plus qu'une fonction objectif ( $n \geq 2$ ), chaque fonction peut avoir un optimum différent. Le but d'un problème multicritères est donc de trouver de bons compromis, plutôt qu'une seule solution, à moins qu'une solution soit optimale pour tous les critères, ce qui est très rarement réalisable. Lorsqu'il y a plusieurs objectifs, la notion d'optimum change, elle sera remplacée par les notions de dominance et d'optimalité de Pareto.

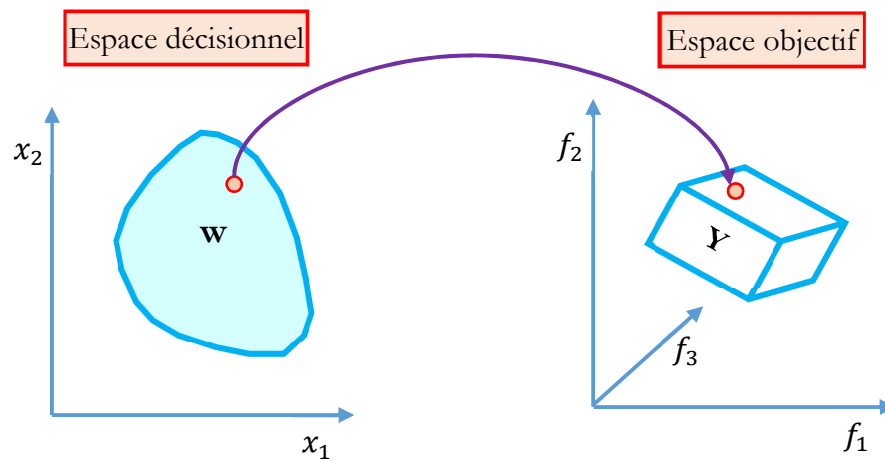


Figure (3.2) : Problème d'optimisation multi-objectif (2 variables de décision et 3 fonctions objectifs)

### 3.2.Approches de résolution de point de vue décideur

La résolution d'un problème d'optimisation multicritère nous donne souvent une multitude de solutions appelées solutions Pareto. L'ensemble de solutions obtenues à la fin de la recherche définit la surface de compromis. C'est après avoir trouvé ces solutions que d'autres difficultés surviennent : il faut sélectionner une solution dans cet ensemble. La solution choisie doit refléter les compromis opérés par le décideur. Pour surmonter ces difficultés, il est nécessaire de faire intervenir l'humain à travers un décideur, pour le choix final de la solution à garder. Ainsi, avant de se lancer dans la résolution d'un problème multicritère, il faut se poser la question du type de méthode d'optimisation à utiliser.

[Irisarri, 2009] distingue trois grandes familles de méthodes selon la manière dont sont combinés les processus de recherche et de prise de décision. Soit, le décideur intervient dès le début de la

définition du problème en exprimant ses préférences afin de transformer le problème multicritère en un problème monocritère, soit il effectue son choix dans l'ensemble des solutions proposées par le solveur multiobjectif, soit il oriente la recherche de façon interactive en fonction de ses préférences, [Clarisse, 2005].

- **Les méthodes d'optimisation a priori** : dans ce cas, le compromis que l'on désire faire entre les objectifs a été défini avant l'exécution de la méthode. Ainsi une seule exécution permettra d'obtenir la solution recherchée. Cette approche est donc rapide, mais il faut cependant prendre en compte le temps de modélisation du compromis et la possibilité pour le décideur de ne pas être satisfait de la solution trouvée et de relancer la recherche avec un autre compromis.
- **Les méthodes d'optimisation progressives** : les processus de décision et d'optimisation sont alternés. Ces méthodes permettent au décideur d'orienter la recherche de façon interactive en fonction de ses préférences et des solutions déjà obtenues. Ce type de démarche nécessite une expertise approfondie du problème considéré ainsi que des outils employés et requiert une adaptation particulière pour chaque problème traité. De plus, l'aspect interactif de la méthode de résolution induit une présence permanente du décideur tout au long de l'optimisation, ce qui est une limite opérationnelle.
- **Les méthodes d'optimisation a posteriori** : dans cette troisième famille de méthodes, on cherche à fournir au décideur un ensemble de bonnes solutions bien réparties. Il peut ensuite, au regard de l'ensemble des solutions, sélectionner celle qui lui semble la plus appropriée. Ainsi, il n'est plus nécessaire de modéliser les préférences du décideur (ce qui peut s'avérer être très difficile), mais il faut en contrepartie fournir un ensemble de solutions bien réparties, ce qui peut également être difficile et requérir un temps de calcul prohibitif (mais ne nécessite pas la présence du décideur).

L'inconvénient qu'il faut souligner est que, maintenant, il faut générer un ensemble de solutions bien réparties. Cette tâche est non seulement difficile, mais, en plus, peut requérir un temps d'exécution prohibitif, [Collette et al. 2002].

Nous nous intéressons dans ce travail à la famille de méthodes a posteriori, où la modélisation des préférences n'est pas requise et le procédé d'optimisation doit être puissant afin de fournir l'ensemble de solutions Pareto optimales ou à défaut une très bonne approximation de la frontière Pareto. Dans ce type de méthodes, deux phases importantes sont à considérer : la première est la phase de recherche de l'ensemble des solutions Pareto optimales, que nous appellerons de façon abusive, résolution du problème d'optimisation et la deuxième est la phase de choix, parmi ces solutions, ceux qui relèvent de l'aide à la décision. Il est à noter que la deuxième phase n'est pas prise en considération dans ce travail.

### 3.3. Hiérarchisation des méthodes d'optimisation multicritère

La plupart des problèmes du monde réel nécessitent l'optimisation simultanée de plusieurs critères souvent antagonistes, incohérents et dépendants les uns des autres. Tandis qu'en optimisation monocritère (ou mono-objectif), la solution optimale est généralement clairement définie, cela n'est pas le cas pour les problèmes à multicritère (ou multi-objectifs). Au lieu d'une solution optimale, il existe plutôt un ensemble de solutions qui sont issu des compromis. Cet ensemble est généralement dénommé (l'ensemble des solutions optimales de Pareto). Les solutions obtenues sont optimales dans le sens qu'aucune autre solution dans l'espace de recherche n'est supérieure à elles, lorsque tous les objectifs sont considérés simultanément.

Dans la littérature, une attention particulière a porté sur les problèmes bi-critères en utilisant les méthodes exactes tels que la méthode Branch and Bound, [Mavrotas et al. 1998], [Mavrotas et al. 2005], [Audrey, 2015], l'algorithme A\*, [Mandow et al. 1996], [Stewart et al. 1991], et la programmation dynamique [Carraway et al. 1990]. Ces méthodes sont efficaces pour des problèmes de petites tailles. Pour des problèmes à plus de deux critères ou de grandes tailles, généralement, il n'existe pas de procédures exactes efficaces, étant donné les difficultés conjuguées de la complexité NP-difficile et le cadre multicritère des problèmes.

Pour élargir l'éventail de résolution, les méthodes heuristiques sont nécessaires pour résoudre les problèmes de grandes tailles et/ou les problèmes avec un nombre de critères supérieur à deux. Néanmoins, ces méthodes ne garantissent pas l'exactitude de l'ensemble des solutions Pareto optimales, mais une approximation de cet ensemble est offerte. Les méthodes heuristiques peuvent être divisées en deux classes : d'une part les algorithmes spécifiques à un problème donné qui utilisent les connaissances du domaine, et d'autre part les algorithmes généraux applicables à une grande variété de problème multi objectif, appelés métaheuristique.

Plusieurs adaptations métaheuristiques ont été proposées dans la littérature pour la résolution de problème multi objectif et la détermination des solutions Pareto : le recuit simulé [Serafini, 1992], [Friesz et al. 1993], [Siarry P., 1999], la recherche Tabou, [Glover, 1986], [Glover, 1990], [Glover et al. 1997], et les algorithmes évolutionnaires : algorithmes génétiques, colonies de fourmis et essaims particuliers. Les méthodes de résolution des problèmes d'optimisation multicritères peuvent être regroupées comme le montre la Figure (3.3).

## 4. Approches de résolution des problèmes d'optimisation multicritères

Les approches utilisées pour la résolution des problèmes d'optimisation multicritère peuvent être classées en deux groupes, Figure (3. 3) :

- **Approches non-Pareto** : ces approches sont classées en deux catégories ; les approches scalaires basées sur la transformation du problème multicritère en problème monocritère, ces approches nécessitent pour le décideur d'avoir une bonne connaissance de son problème, et les approches non-scalaires, qui gardent l'aspect multicritères du problème, mais utilisent des opérateurs de recherche qui traitent séparément chacun des objectifs.
- **Approches Pareto** : Les approches Pareto utilisent directement la notion d'optimalité Pareto dans leur processus de recherche. Le processus de sélection des solutions générées est basé sur la notion de non-dominance.

Plusieurs ouvrages et articles de synthèse plus complets sur ces approches peuvent être consultés notamment dans [Ulungu et al. 1994], [Miettinen, 1999], [Ehrgott, 2000], [Ehrgott et al. 2000], [Deb, 2001], [Collette et al. 2002] et [Mammeri, 2013].

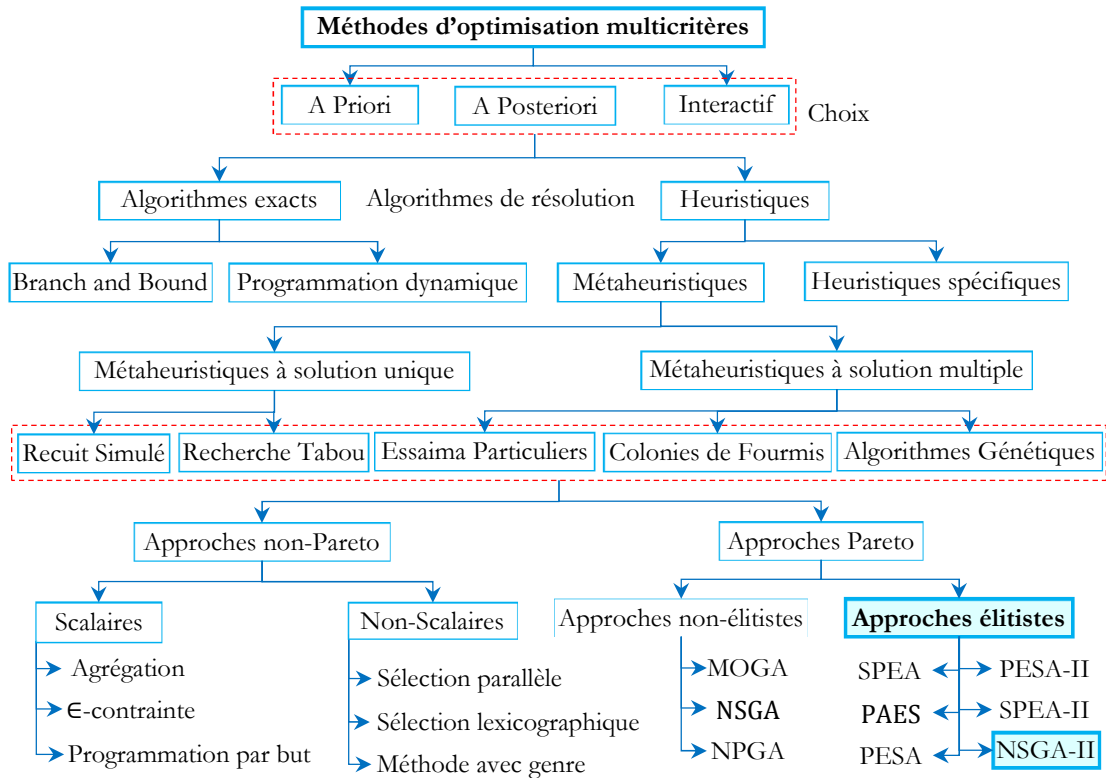


Figure (3. 4) : Classification générale des méthodes d'optimisation multicritère

### 4.1. Approches non Pareto

Les approches non Pareto sont classées en deux catégories : les approches scalaires, qui transforment le problème multiobjectif en problème monocritère et les approches non scalaires, qui gardent l'approche multicritère, mais en traitant séparément chacun des critères (objectifs). Ces approches sont sensibles au paysage du front de Pareto (convexité, continuité, ...), et trouvent souvent



de difficultés à trouver les solutions de compromis puisqu'elles se focalisent sur les portions extrêmes du front. Elles sont efficaces et faciles à implémenter seulement pour les problèmes d'optimisation multicritères avec un nombre réduit d'objectifs.

#### 4.1.1. Approches scalaires

Le principe des approches scalaires est de revenir à un problème monocritère via un ensemble de paramètres (poids ou contraintes sur les objectifs). Parmi ces approches nous citons : les Méthodes d'agrégation par pondération, Méthode  $\epsilon$ -contraintes, Méthode de but à atteindre, Méthode de min-max, Méthode de Goal programming et la méthode de Benson.

Nous présentons ici deux méthodes scalaires : la méthode d'agrégation par pondération et la méthode  $\epsilon$ -contrainte, [Haines et al. 1971], [Chankong et al. 1983], [Jozefowicz, 2013].

##### 4.1.1.1. Agrégation par pondération

Cette méthode de résolution est la plus évidente et probablement, la plus utilisée en pratique. Elle consiste à ramener le problème multicritère à un problème monocritère sous forme de la somme d'une combinaison linéaire des objectifs initiaux. Ainsi, il s'agit d'associer à chaque critère un coefficient de pondération. Le problème d'optimisation, se transforme alors de la manière suivante :

$$\min \left( \sum_{1 \leq i \leq m} w_i f_i(x) \right) \quad (3.4)$$

Où les poids  $w_i \geq 0$  sont tels que  $(\sum_{1 \leq i \leq m} w_i = 1)$ , et  $m \geq 2$ .

En réalité Fixer un vecteur poids revient à trouver un hyper-plan dans l'espace objectif (une droite pour un problème bi-critère) avec une orientation fixée.

La solution Pareto optimale est le point où l'hyperplan possède une tangente commune avec l'espace réalisable ; le point  $(x)$  dans la Figure (3.4-a). Donc, pour une agrégation donnée, il n'existe en général qu'une seule valeur optimale. Ces approches résolvent le problème en utilisant différentes valeurs pour les poids qui fournissent différentes solutions supportées. Mais, dans le cas d'une frontière Pareto concave, les solutions non supportées sont alors négligées. La Figure (3.4-b) représente un cas où deux solutions Pareto optimales peuvent être trouvées.

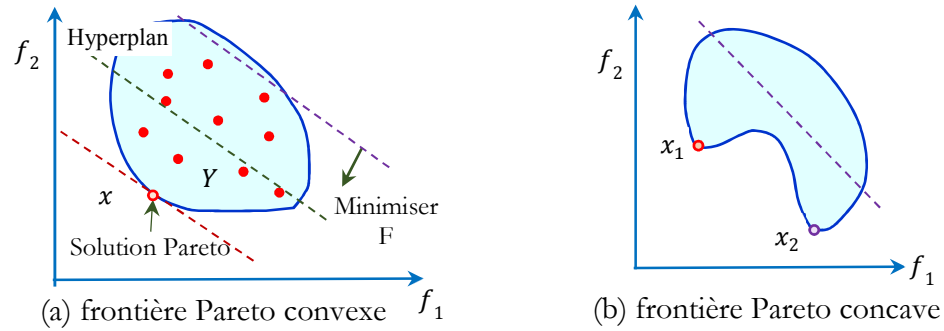


Figure (3.5) : Approche d'agrégation

Si, le domaine réalisable est convexe, cette méthode ne permet pas de trouver les solutions enfermées dans une concavité (les solutions non supportées). Les résultats obtenus avec de telles méthodes dépendent fortement des valeurs affectées au vecteurs poids  $w_i$ . En effet, ces derniers doivent être choisis en fonction des préférences du décideur (cette tâche est délicate).

#### 4.1.1.2. Méthode $\epsilon$ -contrainte

Méthode d'optimisation  $\epsilon$ -contrainte consiste à retenir l'un des critères  $f_k$  comme critère unique à optimiser, et de transformer les autres en contraintes. Le problème d'optimisation peut être donc reformulé sous la forme suivante :

$$\min_{1 \leq k \leq m} f_k(x) \quad (3.5)$$

Avec  $f_j(x) \leq \epsilon_j, j = 1, 2, \dots, m, \text{ et } j \neq k \text{ et } \epsilon = (\epsilon_1, \dots, \epsilon_{k-1}, \epsilon_{k+1}, \dots, \epsilon_m),$

La connaissance a priori des intervalles appropriés pour les valeurs de  $\epsilon$  est exigée pour tous les critères, [Berro, 2001]. La méthode du  $\epsilon$ -Contrainte est facile à mettre en place. Elle exige de fixer un seuil de performance sur n-1 critères. Le principal inconvénient de cette méthode est le fait qu'elle soit gourmande en temps de calcul s'il y a trop de contraintes, [Jeong, 2011].

Les différentes méthodes de l'approche scalaire transforment un problème d'optimisation multicritère en un ou plusieurs problèmes monocritère. Cette transformation permet d'utiliser facilement les méthodes d'optimisation monocritère. Certaines de ces approches ne peuvent pas résoudre complètement des problèmes non convexes et elles sont très sensibles à la forme du front Pareto. D'autres, même si elles peuvent traiter les problèmes non convexes, elles restent quand même sensibles à la forme du front Pareto. Ces méthodes nécessitent souvent une bonne connaissance du problème a priori, notamment pour fixer les vecteurs de poids ou les points de référence.

#### 4.1.2. Approches non-scalaires

Ces approches sont de type a posteriori. Contrairement aux approches scalaires, elles ne transforment pas le problème multicritère en un problème monocritère, mais utilisent des opérateurs

spécifiques qui traitent séparément les différents objectifs [Ehrgott, 2005], elles n'utilisent pas non plus la notion de dominance Pareto. Il existe trois types de méthodes.

#### 4.1.2.1. La méthode de sélection parallèle

L'approche VEGA (Vector Evaluated Genetic Algorithm) a été proposée par [Schaffer, 1985]. Elle est la première approche qui adapte un algorithme génétique pour la résolution de problèmes multiobjectifs. VEGA est fondée sur une idée simple : Si nous avons  $k$  objectifs et une population de  $n$  individus, une sélection de  $n/k$  individus est effectuée pour chaque objectif. Ainsi  $k$  sous-populations vont être créées, chacune d'entre elles contenant les  $n/k$  meilleurs individus pour un objectif particulier. Les  $k$  sous-populations sont ensuite mélangées afin d'obtenir une nouvelle population de taille  $n$ . Le processus se termine par l'application des opérateurs génétiques de modification (croisement et mutation). Il est à noter que de nombreuses variations autour de cette technique ont été effectuées, à savoir : paramètre pour contrôler le taux de sélection, [Ritzel, 1994], intégration de VEGA avec dominance de Pareto, [Tanaki, 1995], application à un problème contraint, [Surry, 1995]. Pour plus de détail voir [Schaffer, 1985], [Berro, 2001], [Figure (3.5)].

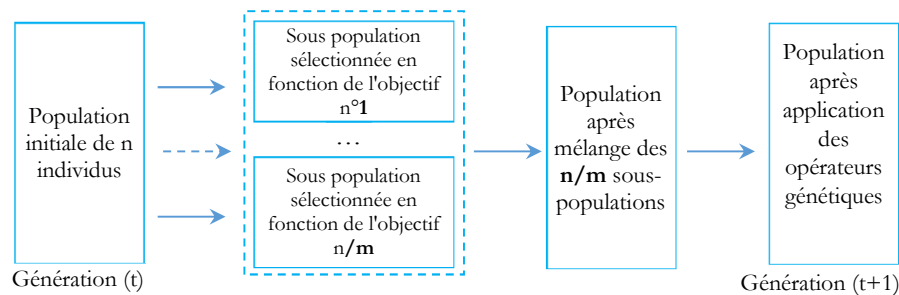


Figure (3.6) : Schéma de fonctionnement de VEGA

VEGA, est populaire, car, facile à implémenter. Elle a tendance à créer des sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. VEGA ignore les solutions compromis et les solutions sur les zones non convexes du front de Pareto ne peuvent être générées.

#### 4.1.2.2. Utilisation des genres

Cette méthode est proposée par [Allenson, 1992]. Allenson a utilisé un algorithme génétique canonique dans lequel un nombre égal d'individus des deux genres sera maintenu. La population est initialisée avec autant de males que de femelles, puis à chaque génération, les enfants remplacent les plus mauvais individus du même genre. La création des enfants s'effectue par croisement mais leur genre est choisi aléatoirement et leur attracteur est créé en fonction de plusieurs heuristiques différentes (aléatoire, clonage ou croisement).

En 1996, Lis et Eiben ont également réalisé un algorithme basé sur l'utilisation des genres, mais dans ce cas l'algorithme n'est pas limité à deux genres, [Lis et al. 1996]. Il peut y avoir autant de genres que d'objectifs du problème. Ils ont également modifié le principe de croisement. Pour générer un enfant, un parent de chaque genre est sélectionné. Ensuite un croisement multipoint est effectué et le parent ayant participé le plus, en nombre de gènes, à l'élaboration de l'enfant transmet son genre. En cas d'égalité le choix s'effectue aléatoirement entre les parents égaux. L'opérateur de mutation effectue un simple changement de genre.

#### **4.1.2.3.La méthode lexicographique**

En 1985 Fourman a proposé une méthode dans laquelle les objectifs sont préalablement rangés par ordre d'importance par le décideur. Ensuite, l'optimum est obtenu en minimisant tout d'abord le critère le plus importante puis le deuxième et ainsi de suite, [Fourman, 1985].

#### **4.1.2.4.La méthode avec genre**

[Allenson, 1992] a proposé une méthode basée sur la notion de genre (masculin ou féminin) et d'attracteur sexuel pour traiter un problème bi-critères. L'exemple proposé consiste à minimiser la longueur d'un pipeline tout en réduisant l'impact écologique de sa construction. En affectant un objectif à chaque genre, l'auteur espère minimiser les deux objectifs simultanément car un genre sera toujours jugé d'après l'objectif qui lui a été assigné, [Hoël, 2011]. L'inconvénient de ce type de méthodes est qu'elles tendent à générer des solutions qui sont largement optimisées pour certains objectifs et très peu pour les autres. Les compromis sont délaissés.

## **4.2.Approches Pareto**

Les approches Pareto sont de type a posteriori. Au 19ème Siècle, Vilfredo Pareto, un mathématicien Italien, formule le concept suivant : dans un problème multiobjectif, il existe un équilibre tel que l'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs. Les approches Pareto utilisent directement la notion de dominance dans la sélection des solutions générées. Le principal avantage de ces approches, c'est l'optimisation simultanée d'objectifs généralement antagonistes. [Goldberg et al. 1989] a été l'un des premiers à utiliser cette notion. Ces approches ont connu un important développement en conjonction avec les algorithmes évolutionnistes à population à partir de la seconde moitié des années 90. Elles sont devenues la principale approche employée pour résoudre les problèmes multicritères du fait de leur capacité à trouver un ensemble potentiellement efficace via la recherche menée sur une population de solutions. De manière générale, les algorithmes évolutionnistes multicritère affectent un score à une solution

selon qu'elle est dominée ou non par d'autres solutions de la population courante et éventuellement si elle domine d'autres solutions.

#### 4.2.1. Notions préliminaires

##### 4.2.1.1. La dominance au sens de Pareto

L'optimisation simultanée de plusieurs critères, représente la principale difficulté du problème d'optimisation multicritères. En effet, il n'existe généralement pas de solution qui minimise (maximise) tous les critères en même temps. La dominance de Pareto a été introduite avec l'idée de distinguer les solutions potentiellement intéressantes. Une solution est potentiellement intéressante lorsqu'on ne peut pas améliorer la performance sur un critère sans dégrader au moins la performance d'un autre. Les solutions ainsi obtenues sont appelées solutions de compromis.

Une solution est jugée intéressante, si elle vérifie la relation de dominance avec les autres solutions, [Collette et al. 2002]. La littérature offre plusieurs relations de dominance entre autres, la dominance, [Othmani, 1998], la dominance au sens de [Ehrgott, 2000], la cône-dominance, [Collette et al. 2002]. Mais la plus célèbre et la plus utilisée est la dominance au sens de Pareto.

##### 4.2.1.1.1. Relation de dominance

En optimisation multicritère lorsque l'on cherche à minimiser plusieurs critères, on dit que le vecteur  $(\vec{x} = (x_1, x_2, \dots, x_n))$  domine le vecteur  $(\vec{y} = (y_1, y_2, \dots, y_n))$  (on note :  $\vec{x} < \vec{y}$  si et seulement si :  $\vec{x}$  est au moins aussi bon que  $\vec{y}$  dans tous les critères et  $\vec{x}$  est strictement meilleur que  $\vec{y}$  dans au moins un critère. Notons que si, la solution  $(\vec{x})$  domine la solution  $(\vec{y})$ , on écrit  $\vec{x} < \vec{y}$ . Autrement dit,  $\vec{x} < \vec{y}$  si et seulement si :

$$\forall i \in \{1, 2, \dots, n\} : f_i(\vec{x}) \leq f_i(\vec{y}) \text{ il existe au moins un } j \in \{1, 2, \dots, n\} \text{ tel que } f_j(\vec{x}) < f_j(\vec{y}) \quad (3.6)$$

Les solutions qui dominent les autres, mais ne se dominent pas entre elles sont appelées solutions optimales au sens de Pareto (ou solutions non dominées). Lorsque l'on applique la définition de la dominance, on peut définir quatre régions auxquelles on peut attribuer des niveaux de préférence. Ces régions sont représentées à la Figure (3.6).

##### 4.2.1.1.2. Optimalité locale au sens de Pareto

Un vecteur de décision  $\vec{x} \in \mathfrak{R}^n$  est dit optimal localement au sens de Pareto, s'il existe un réel  $\delta > 0$  tel qu'il n'y ait pas de vecteur  $\vec{y}$  qui domine le vecteur  $\vec{x}$  avec  $\vec{y} \in \mathfrak{R}^n \cap B(\vec{x}, \delta)$ , ou  $B(\vec{x}, \delta)$  représente une boule de centre  $\vec{x}$  et de rayon  $\delta$ , Figure (3.7).

##### 4.2.1.1.3. Optimalité globale au sens de Pareto

Un vecteur  $\vec{x} \in \mathfrak{R}^n$  est optimal globalement au sens de Pareto (ou optimal au sens de Pareto) s'il n'existe pas de vecteur  $\vec{y} \in \mathfrak{R}^n$  tel que  $\vec{y}$  domine le vecteur  $\vec{x}$ , Figure (3.8).

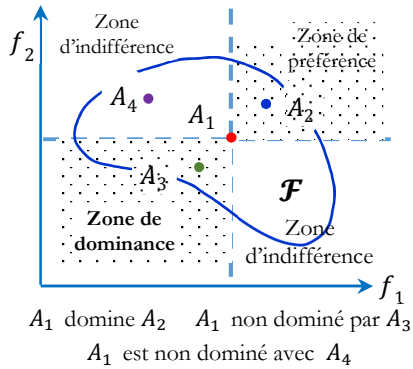


Figure (3.7) : Les niveaux de préférence dans la relation de dominance

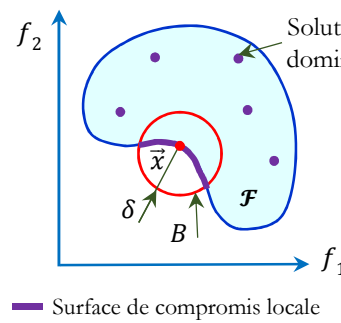


Figure (3.8) : Optimalité locale au sens de Pareto

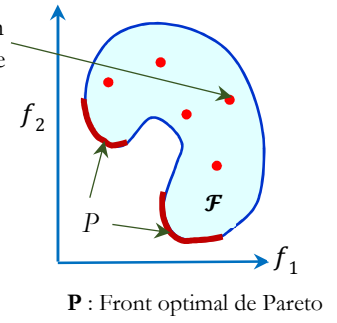


Figure (3.9) : Optimalité globale au sens de Pareto

L'ensemble P des solutions Pareto-optimales globales regroupe tous les meilleurs compromis possibles au problème multicritère. Cet ensemble est aussi qualifié de front optimal de Pareto, ou plus généralement de surface optimale de Pareto, pour des problèmes de dimension plus élevée.

#### 4.2.1.2. Front de Pareto

Le front de Pareto est l'ensemble des solutions de compromis équation (3.7). Sur la Figure (3,9), A et B sont deux points du front de Pareto : A ne domine pas B et B ne domine pas A, mais tous les deux dominent le point C. Le but de l'optimisation multiobjectif est de déterminer le front de Pareto pour un problème donné. Par extension, et comme le montre la figure (3.10), l'ensemble des solutions non dominées de l'espace des solutions peut être considéré comme un front de Pareto local.

$$FP^* = \{\vec{x} \in \mathcal{F} \mid \nexists \vec{y} \in \mathcal{F} \quad \vec{x} < \vec{y}\} \quad (3.7)$$

#### 4.2.1.3. Point idéal et le point Nadir

Le point idéal est utilisé dans beaucoup de méthodes d'optimisation comme point de référence. Ses coordonnées sont obtenues en optimisant chaque critère séparément. Le point nadir, lui, sert à restreindre l'espace de recherche ; il est utilisé dans certaines méthodes d'optimisation interactives. Ses coordonnées correspondent aux pires valeurs obtenues par chaque critère lorsque l'on restreint l'espace des solutions à la surface de compromis. Les deux points idéaux et nadir sont illustrés à la Figure (3.11).

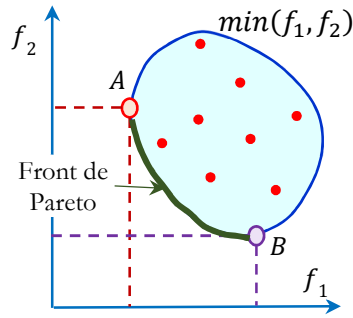


Figure (3.10) : Front de Pareto

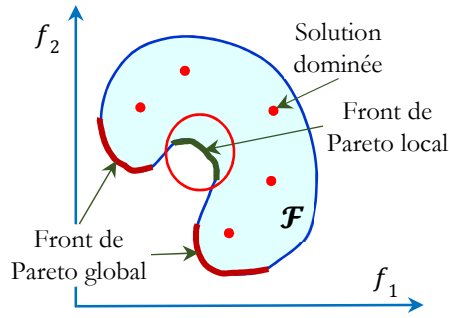


Figure (3.11) : Optimalité globale au sens de Pareto

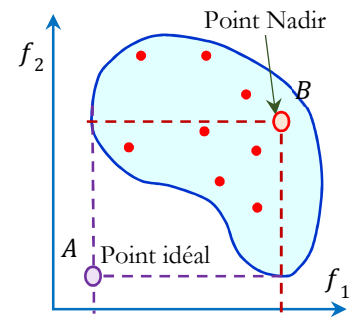


Figure (3.12) : Représentation du point idéal et du point Nadir.

#### 4.2.2. L'équilibre souhaité entre l'intensification et la diversification

Deux objectifs doivent être pris en compte dans la résolution d'un problème d'optimisation multicritères, le premier, c'est l'intensification (exploitation) qui indique la convergence vers la frontière Pareto. Le deuxième, est la diversification (exploration), qui trouve les solutions diversifiées le long de la frontière Pareto.

La Figure (3.12) illustre les différentes qualités de solution, où la (a) présente une solution de bonne qualité en termes de convergence mais mauvaise pour la diversité. La (b) montre la solution de bonne diversité mais de mauvaise qualité pour la convergence. Par contre la (c) présente la disposition idéale des solutions (l'équilibre souhaité).

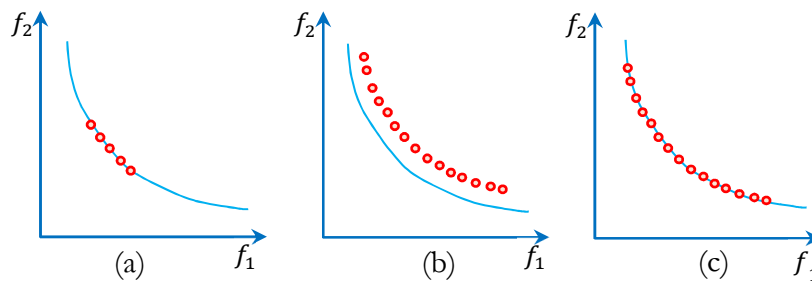


Figure (3.13) : Illustration de la convergence et la diversité en

#### 4.2.3. Mécanisme de sélection Pareto (Ranking)

Un des principaux avantages des algorithmes évolutionnaires pour l'optimisation multicritère est qu'ils permettent non seulement la mise en œuvre d'approches non Pareto, mais aussi l'implémentation d'approches Pareto. En effet, ces algorithmes nécessitent de pouvoir associer une valeur scalaire unique (la fitness), au vecteur des objectifs. Ce principe appelé ranking, consiste à classer les individus en leur donnant un rang. La valeur d'adaptation est alors attribuée à chaque individu en se basant sur son rang. Cette fitness sera utilisée dans l'étape de sélection de l'algorithme (c'est ce

mécanisme de sélection Pareto qui offre une alternative élégante et efficace aux algorithmes évolutionnaires de s'adapter facilement au cas multiobjectif).

Plusieurs méthodes de ranking ont été utilisées dans la littérature (NSGA, [Srinivas, 1995], NDS, [Fonseca, 1995]. Cet ordre dépend de la notion de dominance et donc directement de l'optimalité Pareto. La méthode de ranking permet ainsi de converger vers les solutions Pareto optimales.

Goldberg présenta dans ses travaux une procédure itérative, pour calculer ce rang : initialement, tous les individus non dominés de la population reçoivent le rang 1 et sont retirés temporairement de la population. Puis, les nouveaux individus non dominés reçoivent le rang 2 avant d'être à leur tour retirés. Le processus s'itère tant qu'il reste des individus dans la population. La valeur d'adaptation de chaque individu correspond à son rang dans la population. Ainsi, l'évaluation d'un individu ne dépend pas uniquement de lui-même, mais aussi de la population. Ce principe a l'avantage de ne pas hiérarchiser les objectifs entre eux. Mais l'augmentation de la taille de l'espace de recherche (surtout créée par l'augmentation du nombre de variables) peut influencer la performance de cette sélection. En effet, comment sélectionner certains individus si tous ont le même rang ! Ce cas peut arriver d'autant plus que le nombre de variables est grand, et que le problème est multimodal, [Barichard, 2003]. La Figure (3.13) et l'algorithme (3.1) représentent l'affectation des rangs de Pareto aux solutions,  $n$  est le nombre de solutions.

```

Assignment du rang de Pareto
Rangcourant = 1
m = n.
Tant que (n ≠ 0) Faire
  Pour i de 1 jusqu'à m Faire
    Si x est non dominé
      Rang (xi, t) = Rang courant
    Fin si
  Fin pour
  Pour i de 1 jusqu'à m Faire
    Si Rang (xi, t) = Rangcourant
      Ranger xi dans la population temporaire
      n = n + 1.
    Fin pour
  Rangcourant = Rangcourant + 1
  m = n.
Fin tant que
    
```

Algorithme (3.1) : Assignment du rang de Pareto

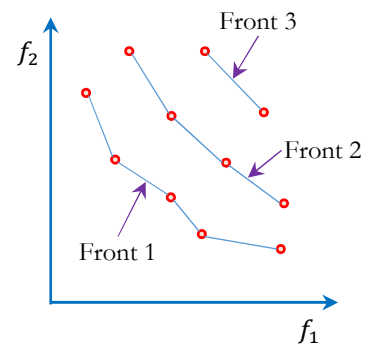


Figure (3.14) : le Ranking

#### 4.2.4. L'élitisme

L'élitisme permet de conserver les meilleurs individus dans les générations futures. L'une des premières implémentations de ce mécanisme dans un algorithme génétique est présentée dans, [De Jong, 1975].



L'élitisme est un mécanisme qui permet la conservation des bonnes solutions lors du passage de la génération actuelle à la génération suivante. L'élitisme améliore d'avantage, les performances des algorithmes et la qualité des solutions. Chose prouvée par les expérimentations de De Jong, où il a observé qu'un algorithme génétique élitiste améliore significativement les résultats sur des fonctions uni-modales. Mais, il peut causer une convergence prématurée sur des fonctions multimodales. Du fait, que Réaliser un algorithme élitiste dans le cadre des problèmes multicritères est plus difficile que pour les problèmes à un objectif. Les approches d'élitisme, [Ishibuchi et al. 1996], [Parks et al. 1998] et [Zitzler et al. 1999] tendent à utiliser une population externe d'individus dans laquelle est stocké le meilleur ensemble des points non dominés collectés jusqu'ici. Cet ensemble est mis à jour continuellement pendant la recherche, et les individus stockés continus à participer avec une certaine probabilité à l'étape de sélection. Ils peuvent ainsi se reproduire et transmettre leurs caractéristiques aux générations suivantes. Cette méthode permet ainsi une intensification de la recherche. Les algorithmes élitistes permettent l'obtention de meilleurs résultats sur un grand nombre de problèmes multicritères [Zitzler et al. 1999], [Deb et al. 2001] et [Barichard, 2003].

#### 4.2.5. Méthodes de diversification

La diversité est une notion déjà importante lors de l'optimisation de problèmes à un objectif, elle devient prépondérante lorsque l'on traite de problèmes multicritères. Maintenir un certain degré de diversité dans la population d'un algorithme évolutionnaire consiste à éviter que la population ne converge prématurément vers une petite zone de l'espace de recherche ou de l'espace des objectifs. En effet, s'il n'existe pas de mécanisme de contrôle de la diversité, les opérations de sélection vont privilégier trop vite certains individus meilleurs à cette étape de la recherche. Cette convergence prématurée a comme effet de limiter la recherche à un sous-ensemble plus restreint de l'espace de recherche, qui peut ne contenir aucune solution optimale. Dans le cas de problèmes multicritères, converger prématurément rend impossible la découverte de l'intégralité du front Pareto.

En effet, les individus de la population se focaliseront sur une partie du front Pareto, et ne se répartiront pas sur la totalité de ce front Pareto. Pour remédier à ce problème, de nombreuses techniques ont été développées. Celles-ci influent sur la pression de sélection, afin de privilégier certains individus, permettant d'obtenir une population plus diversifiée. Cependant, ces techniques ajoutent un coût calculatoire non négligeable pour l'algorithme, elles doivent donc être choisies avec soin. Nous allons maintenant présenter les mécanismes de diversification les plus couramment intégrés aux algorithmes évolutionnaires.

#### 4.2.5.1. Nichage

L'opérateur de sélection dans les algorithmes évolutionnaires duplique, d'une génération à l'autre, les individus les mieux adaptés et conduit nécessairement à un phénomène de dérive génétique, c'est à dire, à terme, tous les individus de la population sont identiques et lorsque, l'algorithme converge vers une solution prématurée non optimale ou locale, alors, les possibilités d'évolution s'en trouvent réduites, car des opérateurs tels que les croisements sont alors incapables de faire apparaître de nouvelles solutions. Ce phénomène est un des points critiques de l'approche évolutionnaire. Évidemment, comme dans la nature, c'est la diversité qui permet l'évolution des populations et réduit le risque de convergence prématurée. Mais en réalité, même si l'algorithme converge convenablement, alors, tous les individus se trouvent concentrés sur l'optimum global. Cette situation peut être trompante, dans la mesure où la présence d'un individu sur l'optimum global est suffisante, le reste de la population pouvant être dédié à l'exploitation d'autres solutions. En effet, les mécanismes de nichage sont conçus en réalité pour réduire la dérive génétique et permettre d'explorer en parallèle plusieurs solutions optimales [Mahfoud, 1995], [Sareni, 1998].

Les techniques de nichage reposent sur une analogie entre les domaines de recherche en optimisation et les écosystèmes naturels. Dans la nature, chaque espèce se distingue par des caractéristiques biologiques propres et évolue de manière à occuper une niche écologique. Dans chaque niche, les ressources naturelles sont limitées et doivent être partagées entre les représentants qui la peuplent, [Benameur, 2010].

Par analogie en optimisation, les niches symbolisent les optima de la fonction (les régions de l'espace présentant des ressources susceptibles d'être exploitées par les différentes espèces) [Goldberg et al. 1987]. Les espèces sont constituées par des groupes d'individus similaires, la similarité étant caractérisée à partir d'un critère de distance, et d'un seuil de dissimilarité (ou de voisinage), qui peut être défini de façon implicite ou explicite, [Sareni, 1998]. Les méthodes de nichage visent à induire l'émergence de niches et d'espèces dans l'espace de recherche, c'est-à-dire à répartir les individus de la population sur les optima de la fonction à optimiser. En principe, les techniques de niche utilisent deux stratégies. La première est basée sur la modification de la structure de certaines régions de l'espace de recherche pour prévenir la convergence de l'algorithme vers ces sections. Cette approche englobe les techniques de surpeuplement, de remplissage ou de partage. La seconde approche impose des contraintes géographiques à la population pour prévenir la prolifération rapide d'individus très performants. Les modèles des îlots et de populations isolées utilisent en général cette seconde stratégie, [El-Imrani, 2000].

Plusieurs méthodes de niche ont été reportées dans la littérature, incluant les méthodes : de partage, [Goldberg et al. 1987] et de sa version améliorée, [El-Imrani et al. 1999], de niche séquentielle, [Beasley et al. (1993), de niche dynamique, [Miller et al. (1996) ou de procédure d'éclaircissement (clearing), [Petrowski, 1996], de surpeuplement (Crowding), [Mahfoud 1994], [Qing et al. 2008]. D'autres méthodes ont été développées utilisant d'autres concepts, telles que les systèmes immunitaires artificiels [De-Castro et al. 2002] et les systèmes basés sur des stratégies financières, [Goldberg et al. 1997].

Plus récemment, le concept de niche écologique a été également étendu à d'autres modèles, à titre d'exemple les essaims particuliers. On peut citer : Nbest PSO, [Brits et al. 2002-1], Niche PSO, [Brits et al. 2002-2], SPSO, [Li, 2004] qui améliore la technique proposée par [Kennedy, 2000], les techniques basées sur des opérations vectorielles, [Schoeman et al. 2005]. Une technique basée sur le concept de nichage séquentiel et la technique d'essaims particuliers PSO (ASNPSO), a été par [Zhang et al. 2006].

#### 4.2.5.2. Fonction de partage - Sharing

Le sharing consiste à modifier la valeur de coût d'un individu (calculée uniquement à partir de la fonction objectif du problème). C'est cette nouvelle valeur qui sera utilisée comme valeur d'adaptation par l'opérateur de sélection. Cette technique, introduite dans Goldberg and Richardson, 1987, est largement utilisée aujourd'hui.

Pour éviter qu'un trop grand nombre d'individus ne se concentrent autour d'un même point, il faut pénaliser la valeur d'adaptation en fonction du nombre d'individus au voisinage du regroupement : plus les individus sont regroupés, plus leur valeur d'adaptation est faible, et des individus proches les uns des autres doivent partager leur valeur d'adaptation. Dans la pratique, on estime ce taux de concentration en ouvrant un domaine autour d'un individu, puis on calcule les distances entre les individus contenus dans ce domaine.

Pour déterminer les bornes du domaine ouvert autour de l'individu choisi, on définit une distance maximale, appelée,  $\sigma_{share}$  au-delà de laquelle les individus ne seront plus considérés comme faisant partie du domaine ouvert. La distance séparant deux individus  $i$  et  $j$  est calculée grâce à la fonction  $d(i, j)$ . La valeur d'adaptation  $F(i)$  d'un individu  $i \in P$  (population) est égale à son coût  $F'(i)$  divisé par sa valeur de niche :

$$F(i) = \frac{F'(i)}{\sum_{j \in P} Sh(d(i, j))} \quad (3.8)$$

où la fonction Sh est définie comme suit :

$$Sh(d(i,j)) = \begin{cases} 1 - \left( \frac{d(i,j)}{\sigma_{share}} \right) & \text{si } d(i,j) < \sigma_{share} \\ 0 & \text{sinon} \end{cases}$$

La fonction distance  $d(i,j)$  peut être définie dans l'espace de recherche, ou dans l'espace objectif par exemple à l'aide d'une distance de Hamming. Ce choix dépend souvent du problème, car le maintien de la diversité dans l'espace objectif, bien qu'il soit souvent plus simple à réaliser, il n'assure pas forcément le maintien de la diversité dans l'espace de recherche.

#### 4.2.5.3. Restriction de voisinage

La restriction de voisinage est une technique proposée par certains travaux, pour maintenir la diversité dans une population. L'idée est de permettre à deux individus de se reproduire s'ils sont similaires. Ceci induit la formation de différentes espèces dans la population, [Fujita et al. 1998]. Par exemple, dans, [Loughlin et al. 1997], les individus sont projetés sur les éléments d'une matrice de dimension  $(n-1)$ ,  $n$  : étant le nombre d'objectifs. Le voisinage est restreint aux individus se trouvant dans un rayon inférieur à  $r$ ,  $r$  étant fixé avant l'exécution de l'algorithme. Certains travaux ont proposé le contraire, c'est-à-dire qu'ils ne permettent pas la reproduction entre individus qui sont similaires, pour prévenir l'inceste, [Eshelman et al. 1991]. Et d'autres approches proposées par [Hajela et al. 1992] utilisent conjointement le principe du partage de fonction et la restriction d'appariement.

#### 4.2.5.4. Crowding

L'approche par crowding consiste à déterminer un représentant par niche découverte. À la différence du sharing, où tous les individus sont susceptibles d'être sélectionnés et de participer aux phases de croisement, mutation et sélection, avec le crowding, seuls les représentants participeront aux différentes étapes de l'algorithme. Le crowding fut introduit par [De Jong, 1975] et fut adapté notamment au travers des travaux de [Blickle, 1996].

#### 4.2.6. Difficultés de maintenir la diversité sur la frontière de Pareto

La difficulté à maintenir une bonne répartition des solutions sur la frontière de Pareto est due principalement à convexité ou non convexité de la frontière de Pareto, à la Continuité et Discontinuité du front de Pareto et au non uniformité de la distribution.

##### 4.2.6.1. Convexité et non convexité

Un front de Pareto  $\mathcal{F}$  est dit convexe si, étant donné deux points distincts quelconques de cet ensemble, le segment qui relie ces deux points est toujours contenu dans l'ensemble  $\mathcal{F}$  des solutions atteignables de l'espace des objectifs, Figure (3.14). La concavité (ou non convexité) d'un espace entraîne l'existence de zones d'optimalité locale pouvant piéger les algorithmes d'optimisation, Figure

(3.15). Certaines méthodes, comme les méthodes agrégatives, ne sont efficaces que lorsque l'espace présente des propriétés de convexité.

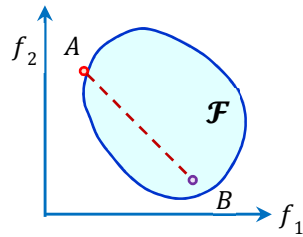


Figure (3.15) : Un ensemble convexe

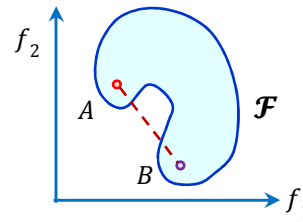


Figure (3.16) : Un ensemble concave

#### 4.2.6.2. Continuité et Discontinuité

La discontinuité d'un front peut être l'un des obstacles qui ralentit la résolution d'un problème multicritère. Cette discontinuité peut être liée à la présence des contraintes Figure (3.16), ou à l'existence de zones optimales locales. Deux exemples de front de Pareto discontinus sont présentés dans la Figure (3.17).

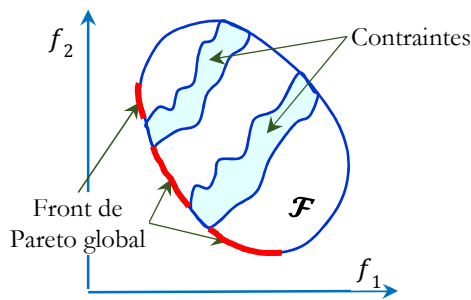


Figure (3.17) : Discontinuité due aux contraintes

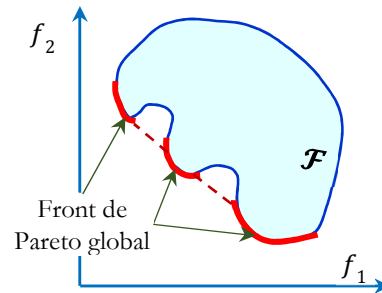


Figure (3.18) : Discontinuité due à la forme du front

#### 4.2.6.3. Non uniformité des solutions

Un problème d'optimisation multicritère peut être caractérisé par une distribution non uniforme des solutions dans l'espace des objectifs. Ainsi, certaines zones de l'espace des objectifs peuvent présenter des densités de solutions plus importantes que d'autres. Si les densités les plus fortes se situent loin du front optimal, les méthodes de résolution vont rencontrer plus de difficultés pour atteindre les solutions globales, [Smairi, 2013].

### 4.3. Difficultés des méthodes d'optimisation multicritère

Plusieurs difficultés probables, peuvent être rencontrées au cours de la résolution d'un problème d'optimisation multicritère. Un processus d'optimisation multimulticritères doit résoudre les deux tâches : guider le processus de recherche vers la frontière de Pareto et maintenir une diversité des solutions pour assurer une bonne répartition sur la frontière de Pareto. L'accomplissement de ces

tâches est très délicat car, les principales difficultés rencontrées dans un problème multicritère sont identiques à celles d'un problème monocritère, mais, elles sont amplifiées par la présence des objectifs interdépendants et souvent antagonistes. En effet, le processus d'optimisation est souvent ralenti ou totalement dérouté par des critères possédant l'une des caractéristiques suivantes : multi-modalité, isolation d'un optimum ou tromperie.

#### 4.3.1. Multi-modalité

La multi-modalité du front de Pareto découle de l'existence de solutions localement Pareto optimales. Les problèmes multimodaux sont caractérisés par l'existence de plusieurs optimums locaux (fronts locaux) successifs, qui sont susceptibles d'empêcher la détermination du front global, Figure (3.18) et Figure (3.19). Dès lors, chaque optimum exerce sur les individus d'une population une attraction différente qui peut piéger le processus de convergence des algorithmes de résolution, [Jérémi, 2003]. Ce problème peut être évité en utilisant une technique de répartition des individus de type sharing ou crowding, [Mahfoud, 1995].

Les algorithmes génétiques sont particulièrement bien adaptés à ce type de problème, de par leur nature distribuée. Les variantes de type (multi-population) exploitent en parallèle plusieurs populations, qui s'attachent à repérer des optima différents, [Yann, 2002].

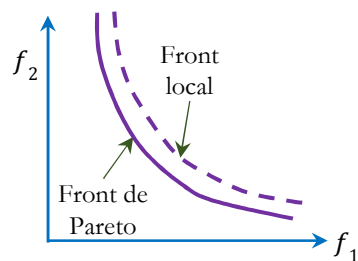


Figure (3.19) : Front Multimodal

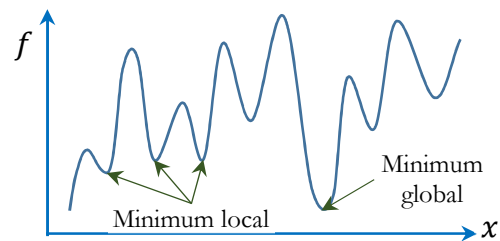


Figure (3.20) : Fonction Multimodal

#### 4.3.2. Isolation d'un optimum

Il existe des problèmes dans lesquels un optimum peut être entouré de grandes zones pratiquement plates. Cet optimum se trouve alors isolé car l'espace de recherche qui l'entoure ne peut pas guider vers lui les individus de la population. Généralement il est très difficile de garantir au décideur l'obtention de ce type d'optimum, (Berro, 2001).

#### 4.3.3. Tromperie

Un problème est trompeur Figure (3.20) lorsqu'il guide la convergence vers une zone non optimale. Nous remarquons que pour les deux problèmes, la probabilité de choix aléatoire d'un point dans une zone sous-optimale est très grande. Dans un premier temps, les méthodes ont une tendance

à converger vers des optima locaux. Par la suite, le processus de mutation entretient ce phénomène de tromperie. Pour éviter cette tromperie, Deb et Goldberg recommandent l'utilisation de techniques de répartition des individus en niches Goldberg et al. (1992). Ils confirment également que le choix d'une taille appropriée de la population est primordial pour éviter la tromperie, Berro (2001).

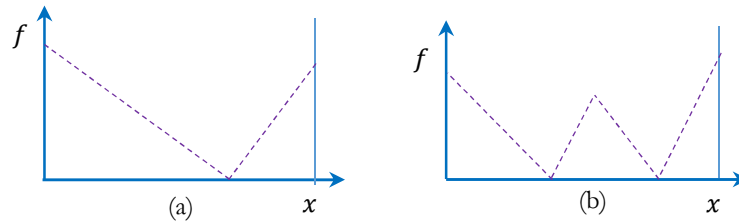


Figure (3.21) : Problèmes trompeurs

#### 4.4. Choix de la stratégie de résolution

La stratégie de résolution à adopter est en fonction du problème à résoudre, et de la connaissance de la structure de sa frontière Pareto. Différentes stratégies peuvent être proposées.

- La structure du problème est bien connue et sa frontière a été montrée convexe : dans ce cas, toutes les solutions Pareto sont supportées. Il est donc possible de rechercher ces solutions en utilisant des agrégations linéaires d'objectifs.
- La structure du problème est bien connue et sans être convexe, il est montré que les solutions supportées sont bien réparties : si l'objectif est d'obtenir un ensemble de solutions représentatif du front Pareto (sans nécessairement obtenir toutes les solutions), alors il est encore possible de ne rechercher que les solutions supportées, à l'aide d'agrégations d'objectifs.
- La structure du problème n'est pas ou mal connue, ou les solutions supportées sont mal réparties sur le front : afin de ne prendre aucune hypothèse sur la répartition des solutions sur le front, il est nécessaire de rechercher à la fois les solutions supportées et non supportées. De plus, dans un environnement non stable, la connaissance de plusieurs solutions Pareto optimales permet de mieux appréhender les aléas en permettant parfois de changer le choix de la solution en fonction d'un changement de condition sans avoir à réaliser une nouvelle recherche de solution.

Les deux premiers cas cités sont bien évidemment les plus favorables. Malheureusement ils nécessitent une bonne connaissance du problème sous étude et en particulier une connaissance de la caractérisation de sa frontière Pareto. Cette caractérisation est très difficile à réaliser car il ne faut pas se focaliser sur quelques instances étudiées, mais bien sur des propriétés générales vérifiées par l'ensemble des instances d'un problème. Ainsi, pour éviter de restreindre à quelques types de problèmes, nous nous placerons dans le cas où la structure du problème est mal connue ou non

favorable. Notre objectif sera alors de rechercher l'ensemble des solutions supportées et non supportées.

#### 4.5.Évaluation des solutions d'optimisation multicritère

L'évaluation de la qualité de ou des solutions d'un problème d'optimisation, est une étape cruciale, après la modélisation, le développement de l'algorithme d'optimisation. La comparaison d'algorithmes dans le cas d'optimisation par des méthodes exacte, comme dans le cas d'optimisation monocritères par des méthodes approchées est triviale. Mais elle n'est pas de même pour l'optimisation multicritère. En effet l'existence d'un ensemble de solutions de compromis et l'absence d'ordre total entre les solutions rendent la mesure de qualité d'un front difficile. Si la notion de dominance au sens de Pareto peut être utilisée pour comparer deux solutions, bien que ces deux solutions puissent être incomparables, la comparaison d'un ensemble de solutions est encore plus délicate. L'évaluation d'un algorithme en termes de qualité des solutions obtenues, nécessite soit de pouvoir évaluer un front (métriques absolues) soit de le comparer de façon quantitative avec les fronts produits par d'autres algorithmes (métriques relatives). Malheureusement cette tâche est délicate puisque la notion de qualité d'un front est elle-même multicritère (intensification, diversification).

En effet, Un ensemble de mesures ont été proposées dans littérature, pour analyser les performances des algorithmes multicritères. Entre autres nous citons : Proportion d'erreur [Veldhuizen, 2000], distance générationnelle [Veldhuizen, 2000], Erreur maximale à la surface de compromis, Mesures évaluant la convergence [Meunier, 2000], Mesures évaluant la diversité [Schott, 1995] et Mesures évaluant la convergence et la diversité [Zitzler, 1999].

Pour une bonne analyse, différentes mesures doivent être utilisées afin de pouvoir analyser à la fois convergence et diversité. L'analyse de performances en multicritères est en elle seule un domaine d'étude encore très ouvert puisqu'il n'existe pas de mesure universellement utilisée. Il est important de retenir qu'aucune des mesures existantes ne peut synthétiser en une seule valeur toute l'information contenue dans la surface de compromis. C'est pourquoi il est souvent nécessaire d'utiliser plusieurs de ces mesures conjointement pour espérer évaluer au mieux la surface de compromis. Le meilleur moyen d'évaluer une surface de compromis, lorsque le problème ne comporte pas plus de trois critères, reste encore l'évaluation graphique.

### 5. Les principales métaheuristiques pour l'optimisation multicritères

Vilfredo Pareto (1848-1923), est un mathématicien Italien, formule le concept suivant : dans un problème multicritère, il existe un équilibre tel que l'on ne peut pas améliorer un objectif sans détériorer au moins un des autres objectifs, [Olivier, 2004]. Ces approches utilisent la notion de



dominance pour comparer les solutions entre elles. L'un des premiers à discuter de l'intérêt de l'utilisation de la notion de dominance pour la recherche de solutions est [Goldberg, 1975]. Ce type de méthodes ne fait subir aucune transformation au problème multicritère ni une préférence pour un objectif par rapport à un autre. Les objectifs sont traités de la même façon et les solutions optimales sont celles qui ne sont pas dominées au sens de Pareto, [Clarisse, 2005]. En général, dans ce type d'approche, une seule exécution suffit pour approcher la frontière Pareto. En plus d'après [Fonseca et al. 1995] même si la nature contradictoire des critères n'est pas prouvée, les métaheuristiques basées sur Pareto peuvent trouver la solution idéale et c'est la meilleure pour tous les critères. Ces approches sont de type a posteriori.

Dans le domaine de l'optimisation multicritères, l'élitisme concerne l'utilisation d'une population secondaire pour maintenir les individus non dominés trouvés durant le processus d'optimisation. Ces algorithmes sont des algorithmes évolutionnaires se basant sur la sélection d'un certain nombre d'individus pour évoluer la population, parfois la sélection se base sur les meilleurs individus, on parle des algorithmes élitistes ou de mauvais individus appelés des algorithmes non élitistes, [Zaghdoud, 2015].

## 5.1. Les techniques Non-élitiste

Les approches que nous allons voir ne conservent pas les individus Pareto optimaux trouvés au cours des itérations, ce qui rend difficile le maintien de la diversité sur la frontière de Pareto. D'où, la convergence des solutions vers la frontière de Pareto est lente, [Berro, (2001)].

### 5.1.1. Multiple Objective Genetic Algorithm (MOGA)

La méthode MOGA, Proposée par [Fonseca et al. 1995] est à la base de la technique de Goldberg, dans laquelle chaque individu de la population est rangé en fonction du nombre d'individus qui le dominant. On utilise une fonction de notation permettant de prendre en compte le rang de l'individu et le nombre d'individus ayant le même rang, [Berro, 2001]. Considérons, pour illustration, un individu  $x_i$  à la génération  $t$ , dominé par  $p_i(t)$  individus de la génération en cours, il se voit attribuer le rang :  $rang(x_i, t) = 1 + p_i(t)$

Tous les individus non dominés sont affectés du rang 1, tandis que les dominés sont pénalisés en fonction de la densité de la population de la région correspondante de la surface de compromis.

**Algorithme MOGA.**

Initialisation de la population  
 Évaluation des objectifs  
 Assignment d'un rang basé sur la dominance  
 Assignment d'une efficacité à partir de ce rang  
**Pour**  $i = 1$  jusqu'à  $N$  **Faire**  
     Sélection aléatoire proportionnelle à l'efficacité ;  
     Croisement ;  
     Mutation ;  
     Évaluation des individus ;  
     Assignment d'un rang basé sur la dominance ;  
     Assignment d'une efficacité à partir de ce rang ;  
**Fin Pour**

Algorithme (3.2) : Algorithme MOGA

L'évaluation de la fitness de chaque individu se fait de la façon suivante :

- 1) Tri de la population selon le rang.
- 2) Attribuer la fitness des individus par interpolation du meilleur (rang 1) au pire (rang  $n \leq N$ ) dans le sens proposé par [Goldberg, 1989], selon une certaine fonction, généralement linéaire.
- 3) faire la moyenne des fitnesses des individus ayant le même rang, de sorte que chacun d'entre eux soit échantillonné à la même vitesse. Ceci permet de maintenir la fitness global de la population, tout en maintenant constante la pression de sélection appropriée.

La méthode MOGA présente le principal, d'être efficace et relativement facile à mettre en œuvre. Néanmoins, elle est critiquée, d'avoir effectué le partage de l'espace des objectifs, ce qui implique que deux vecteurs différents avec les mêmes valeurs de la fonction objectif ne peuvent exister simultanément dans la population. La principale faiblesse de MOGA est que, comme toutes les autres techniques de classement de Pareto, sa performance dépend fortement d'une sélection appropriée du facteur de partage (sharing), [Zidani, 2013].

### 5.1.2. Non dominated Sorting Genetic Algorithm (NSGA)

Dans l'algorithme NSGA proposé par [Srinivas et al. 1993], le calcul de fitness s'effectue en divisant d'abord la population en plusieurs fronts en fonction du degré de dominance au sens de Pareto de chaque individu. Les individus non dominés de la population courante constituent le premier front de Pareto. On attribue alors à tous les individus de ce front la même valeur de fitness factice. Cette valeur est supposée, elle donne une chance égale de reproduction à tous ces individus. Mais pour maintenir la diversité de la population, il est nécessaire d'appliquer une fonction de partage sur cette valeur. Ensuite, ce premier groupe d'individus est temporairement supprimé de la population. On recommence cette procédure jusqu'à l'identification des solutions du deuxième front. La valeur factice de fitness attribuée à ce second groupe est inférieure au plus petit fitness, après application de la

fonction de partage sur le premier front. Ce mécanisme est répété jusqu'à ce que l'on ait traité tous les individus. L'algorithme se déroule ensuite comme un algorithme génétique standard. [Srinivas et al. 1993] utilisent une sélection basée sur le reste stochastique. Mais leur méthode peut être utilisée avec d'autres heuristiques de sélections (tournoi, roulette pipée, etc.). Grâce à sa procédure d'assignement de fitness basée à la fois sur la notion de dominance et la fonction de partage, le NSGA semble le plus approprié à maintenir la diversité de la population et à répartir plus efficacement les solutions sur le front de Pareto. Néanmoins, cet algorithme présente quelques insuffisances en raison de sa complexité de calcul et de sa sensibilité au choix de la valeur  $\sigma_{share}$ , Figure (3.21).

Cette méthode paraît moins efficace en temps de calcul que la méthode MOGA car le temps de calcul de la notation (tri de la population et sharing) est important. Mais l'utilisation d'un sharing sur l'espace d'état et le tri des solutions en différentes frontières semblent plus appropriés à maintenir une grande diversité de la population et à répartir plus efficacement les solutions sur la frontière de Pareto, [Bagchi, 1999]. De plus, cette méthode est applicable dans des problèmes avec un nombre quelconque d'objectifs. Trois critiques ont été soulevées pour cette méthode :

- Sa complexité de calcul de  $O(k.N^3)$  avec  $k$  le nombre d'objectifs et  $N$  la taille de la population, essentiellement due au processus de tri de la population et d'application de l'heuristique de partage.
- Son approche non élitiste : Le tri de la population est une heuristique intéressante pour distribuer la population sur l'ensemble de la frontière de Pareto mais cette procédure ralentit le processus de convergence de l'algorithme. De plus, cet effet est accentué par l'utilisation de la méthode de sélection par reste stochastique.
- La nécessité de spécifier un paramètre de sharing.

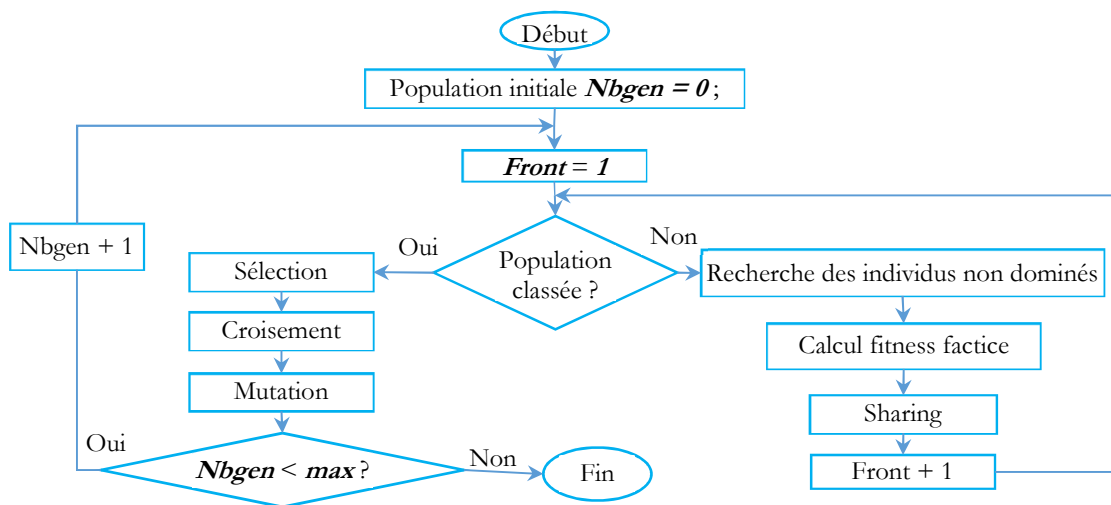


Figure (3.22) : Schéma de fonctionnement de NSGA

### 5.1.3. Niche Pareto Genetic Algorithm (NPGA)

La méthode NPGA proposée par Jeffrey Horn et ses collègues, [Jeffrey et al. 1993], [Horn et al. 1994] utilisent un tournoi basé sur la notion de dominance de Pareto. Elle compare deux individus choisis au hasard, à une sous-population, également choisie au hasard, en général, d'environ 10% de l'ensemble la population. Si l'un d'eux est dominé par les individus sous-population et l'autre non, alors il est positionné dans la population suivante. Dans les autres cas une fonction de sharing est appliquée pour sélectionner l'individu. Le réglage de la taille de la sous population permet d'exercer une pression variable sur la population et ainsi d'augmenter ou de diminuer la convergence de l'algorithme. Le choix de cette taille à un effet majeur sur la vitesse de convergence, une valeur de l'ordre de 10% de N (Nombre d'individus de la population) donne une bonne distribution des individus. Cette valeur apporte un bon compromis entre élitisme et représentativité statistique de la population. Elle permet non seulement d'exercer une pression suffisante sur la sélection des individus, mais également d'éviter une convergence non désirée due à une sélection effectuée sur un échantillon non représentatif de la population. À travers leurs expérimentations Horn et Nafliotis établissent le constat suivant :

- Si la taille de la sous-population est enverront 1% de N, il y a trop de solutions dominées. Cette valeur n'apporte pas assez d'élitisme dans la sélection, d'où un grand nombre de solutions non dominées qui subsistent dans la population.
- Si la taille de la sous-population est enverront 10% de N, une bonne distribution des individus est obtenue. Cette valeur apporte un bon compromis entre élitisme et représentativité statistique de la population. Elle permet non seulement d'exercer une pression suffisante sur la sélection des individus, mais également d'éviter une convergence non désirée due à une sélection effectuée sur un échantillon non représentatif de la population.
- Si la taille de la sous-population est enverront supérieur à 20% de N, il y a une convergence prématurée, car la pression est trop importante lors de la sélection.

Cette méthode nécessite la maîtrise de deux paramètres : le sharing et la taille de sous population. Les solutions trouvées sont de bonne précision et cette approche est plus rapide que l'approche précédente car le sharing n'est appliqué que sur une portion de la population, Algorithme (3.3).

## Algorithme de la fonction de sélection

```

sous_pop () ← Sélection aléatoire de Tspop + 2
candidat1 ← sous_pop (1)
candidat2 ← sous_pop(1)
cand_dominé1 ← faux
cand_dominé2 ← faux
Pour i de 3 à tdom + 2 faire
    cand_comp ← sous_pop (i)
    Si (cand_comp domine cand_dominé1) alors
        cand_dominé1 ← vrai
    Fin Si
    Si (cand_comp domine cand_dominé2) alors
        cand_dominé2 ← vrai
    Fin Si
Fin Pour
Si (cand_dominé1 et non cand_dominé2) alors
    retourne cand_dominé2
sinon
    si (non cand_dominé1 et cand_dominé2) alors
        retourne cand_dominé1
    sinon appliquer l'heuristique de sharing
    Fin Si
Fin Si

```

Algorithme (3.3) : Algorithme de la fonction de sélection

Avec Tspop désigne la taille de la sous-population.

## 5.2. Les techniques élitistes

Dans le domaine de l'optimisation multicritères, la sélection élitiste consiste à maintenir une seconde population appelée archive, contenant les solutions non dominées trouvées au cours des différentes générations passées de l'algorithme évolutionniste. Les individus de cette population participent avec une certaine probabilité à l'étape de sélection et donc à la reproduction de nouveaux individus. En outre les techniques de regroupements ou clustering, niching, et grid-based sont employées pour répartir efficacement les solutions sur la frontière de Pareto et limiter la taille de cette archive. Cette technique est utilisée pour surmonter les difficultés des méthodes non-élitistes. Les paragraphes suivants présentent les différentes évolutions des méthodes élitistes.

### 5.2.1. Strength Pareto Evolutionary Algorithm (SPEA)

Zitzler et Thiele ont proposé en 1998 la méthode élitiste, Strength Pareto Evolutionary Algorithm (SPEA), [Zitzler, 1998]. SPEA est fondé sur les concepts de Pareto. L'élitisme est introduit par le maintien explicite d'une archive (population externe). Cette archive est destinée à contenir un nombre limité de solutions non dominées trouvées collecté depuis le début de l'évolution. Les individus non-dominés de chaque itération sont comparés à ceux de la population externe au sens de

dominance et seules les solutions non dominées sont maintenues dans la nouvelle archive externe, Figure (3.22), Algorithme (3.4).

Il est important de noter que SPEA non seulement elle préserve les individus élites mais aussi les fait participer aux opérations génétiques. Une méthode de clustering est exploitée pour réduire l'ensemble de Pareto sans supprimer ses caractéristiques. Ainsi, qu'une nouvelle méthode de niche, basée sur Pareto, est utilisée afin de préserver la diversité.

L'avantage essentiel de SPEA est qu'elle n'exige pas de réglage de paramètre de sharing.

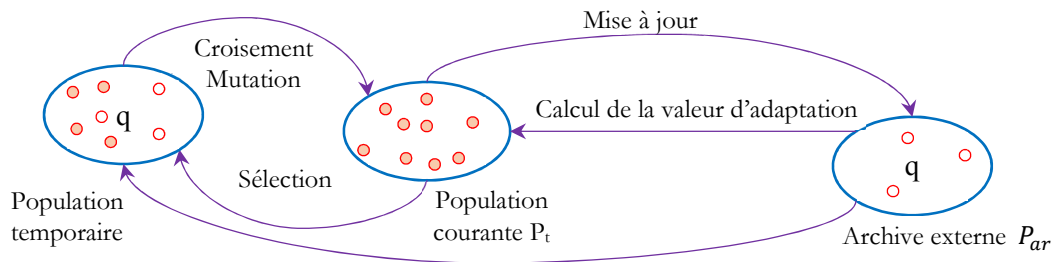


Figure (3.23) : Fonctionnement général de l'algorithme SPEA

Strength Pareto Evolutionary Algorithm (SPEA).
Initialiser la population $P$ et créer l'archive externe vide $P_{ar} = \phi$ ;
Mise à jour de $P$ à partir des individus non dominés de $P_0$
<b>Tant que</b> critère d'arrêt non rencontré <b>faire</b>
Calcul de la valeur d'adaptation pour tous les individus de $P + P_{ar}$
Sélection dans $P_t + P_{ar}$ en fonction de la valeur d'adaptation
Croisement
Mutation
Mise à jour de $P_{ar}$ à partir des individus non dominés de $P$
<b>Fin Tant que</b>

Algorithme (3.4) : Strength Pareto Evolutionary Algorithm

Pour diminuer les chances de sélection des individus ayant une moins bonne évaluation, ou des individus issus de niches déjà importantes un calcul d'adaptation de SPEA est fait.

À la différence des techniques classiques de sharing qui sont basées sur la notion de distance euclidienne dans l'espace des objectifs, SPEA utilise la notion de dominance pour détecter les niches d'individus. En effet, une valeur, appelée valeur de dureté ( $S$ ), est affectée aux individus de l'archive externe  $P_{ar}$ . Cette valeur est déduite à partir du nombre d'individus  $n_i$  qu'un élément  $i \in P_{ar}$  domine faiblement dans la population courante  $P_t$  : Soit  $n_i = |\{i | i \in P_t \wedge f(i) \leq f(j)\}|$ , alors :

$$S_i = \frac{n_i}{|P_t| + 1} \quad (3.9)$$

La division par  $(|P_t| + 1)$  garantit que les valeurs de  $S_i$  sont inférieures à 1. De plus la solution externe dominant moins de solutions de la population courante reçoit une valeur de performance plus petite. La valeur d'adaptation d'un individu  $j$  de  $P_i$  est égale à sa valeur de dureté (strength):  $F_j = S_j$ . Pour un individu  $j$  de la population courante  $P_t$ , la valeur d'adaptation est calculée en réalisant la somme des valeurs de dureté de toutes les solutions externes dominantes :

$$F_j = 1 + \sum_{i \in P_{ar_t} \wedge i \preceq j} S_i \quad (3.10)$$

Au total de la somme est ajouté 1 pour éviter que des individus de  $P$  aient une valeur d'adaptation plus grande que certains individus de  $P_{ar}$ . Il est à noter que, dans ce calcul, une plus petite valeur d'adaptation conduit à une plus grande chance de sélection de l'individu. Ainsi, plus un individu est dominé par les individus de  $P_{ar}$  plus sa valeur d'adaptation décroît, diminuant donc ses chances d'être sélectionné. La sélection par tournoi de la taille 2 effectuée sur l'ensemble  $P_{ar} \cup P_i$  se base sur les valeurs  $S_i$  et  $F_j$  de façon à préférer les valeurs les plus petites.

Les étapes d'une itération SPEA sont organisées comme suit :

- 1) Trouver l'ensemble des individus non-dominés  $\mathcal{F}_1(P_t)$  de la population courante  $P_t$
- 2)  $P_{ar_t} \leftarrow P_{ar_t} \cup \mathcal{F}_1(P_t)$
- 3) Trouver toutes les solutions non-dominées  $\mathcal{F}_1(P_{ar_t})$  de la population externe modifiée  $P_{ar_t}$  et éliminer toutes les solutions dominées :  $P_{ar_t} \leftarrow \mathcal{F}_1(P_{ar_t})$
- 4) Si  $P_{ar_t} > N_{ar}$ , utiliser la technique de clustering pour réduire la taille de l'archive jusqu'à  $N_{ar}$ .  
La population résultante est la population externe de la génération suivante  $P_{ar_{t+1}}$ .
- 5) Calculer la performance des solutions de  $P_{ar_{t+1}}$  utilisant l'équation (3.9) et des solutions de  $P_{t+1}$  selon l'équation (3.10).
- 6) Appliquer sélection par le tournoi de la taille 2, le croisement et la mutation aux individus de l'ensemble  $P_{ar_{t+1}} \cup P_t$  pour créer une nouvelle population  $P_{t+1}$  de taille  $N$ .

### 5.2.2. Méthode Pareto Archived Evolution Strategy (PAES)

La méthode PAES, a été développée et mise en œuvre par Knowles et al. (2000) est une approche assez simple, comme indiqué dans Algorithme (3.5). Son principe n'est pas basé sur une population car elle n'utilise qu'un seul individu à la fois pour la recherche des solutions. Elle utilise une population annexe de taille déterminée permettant de stocker les solutions temporairement Pareto-optimales. Cette archive est utilisée comme un ensemble de références à laquelle chaque individu muté est comparé. PAES utilise également une nouvelle approche pour maintenir la diversité,

qui consiste en une procédure de regroupement qui divise l'espace objectif de manière récursive. En outre, la procédure a une complexité de calcul plus faible que les méthodes traditionnelles de nichage. L'algorithme de PAES se présente en trois parties : la génération d'une solution candidate, la fonction d'acceptation de la solution candidate et l'archivage des solutions non dominées. La méthode de génération d'un nouveau candidat ressemble à la méthode de Hill-Climbing. À chaque itération un nouveau candidat est produit par mutation aléatoire.

L'algorithme de PAES
1-Génération aléatoire d'une solution $c$ et ajout de $c$ à l'archive. 2-Production d'une solution $m$ par mutation de $c$ et évaluation de $m$ . 3- <b>Si</b> ( $c$ domine $m$ ) <b>alors</b> on écarte $m$ . <b>sinon si</b> ( $m$ domine $c$ ) <b>alors</b> on remplace $c$ par $m$ et ajout de $m$ à l'archive. <b>sinon si</b> ( $m$ est dominé par un membre de l'archive) <b>alors</b> on écarte $m$ . <b>sinon</b> on applique une fonction de test ( $c, m, \text{archive}$ ) qui détermine la nouvelle solution courante. 4-On recommence en 3.

Algorithme (3.5) : Algorithme de PAES

La fonction de test est donnée par Algorithme (3.6)

Algorithme de la Fonction de test ( $c, m, \text{archive}$ )
1- <b>Si</b> l'archive n'est pas pleine <b>alors</b> ajout de $m$ à l'archive. <b>sinon si</b> ( $m$ est dans une région moins encombrée qu'une solution $x$ appartient à l'archive) <b>alors</b> ajout de $m$ et suppression d'un membre de la zone la plus encombrée de l'archive. 2- <b>Si</b> ( $m$ est dans une région moins encombrée que $c$ ) <b>alors</b> $m$ est acceptée comme solution courante. <b>sinon</b> on conserve $c$ comme solution courante.

Algorithme (3.6) : Algorithme de la Fonction de test ( $c, m, \text{archive}$ )

Cette méthode est relativement simple à mettre en œuvre. De plus, n'étant pas basée sur un algorithme génétique, elle évite à l'utilisateur le réglage de tous les paramètres de celui-ci. Mais son efficacité va dépendre du choix d'un nouveau paramètre : le paramètre de discrétisation de l'espace des objectifs. La technique de crowding utilisée dans PAES permet une mise à jour de l'archive plus rapide lors des dépassements de capacité que celle de SPEA.

### 5.2.3. Pareto Envelope based Selection Algorithm (PESA)

La méthode PESA a été proposée également par [Knowles et al. 2000]. Elle reprend approximativement le principe de crowding de PAES et définit un paramètre appelé `squeeze_factor` qui représente la mesure d'encombrement d'une zone de l'espace. Alors que PAES est basée sur une



stratégie d'évolution, PESA est une méthode basée sur les algorithmes génétiques. PESA définit deux paramètres concernant la taille des populations d'individus :  $P_I$  (taille de la population interne) et  $P_E$  (taille de la population externe ou archive). Figure (3.23), Algorithme (3.7).

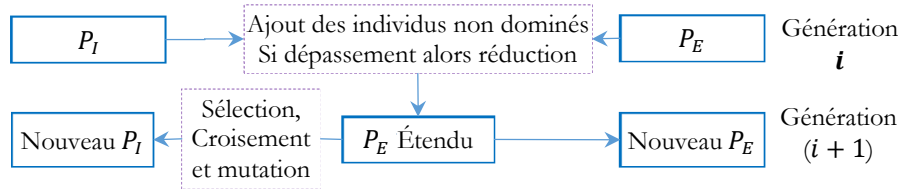


Figure (3.24) : Schéma de fonctionnement de PESA

L'algorithme de PESA
1-Génération aléatoire et évaluation de la population $P_I$ , et initialisation de $P_E$ ( $P_E = \emptyset$ ).
2-Transfert de tous les individus non dominés de $P$
3-Si le critère d'arrêt est réalisé
<b>alors</b> on retourne $P_E$ comme ensemble de solutions.
<b>sinon</b> on supprime tous les individus de $P_I$ et on recrée $P_I$ de la façon suivante :
On sélectionne deux parents dans $P_E$ avec la probabilité $p_c$ par croisement puis on le mute. Avec la probabilité $(1 - p_c)$
On sélectionne un parent et on le mute pour produire un autre enfant.
4-On recommence au 2.

Algorithme (3.7) : Pareto Envelope based Selection Algorithm

Une solution courante de  $P_I$  peut entrer dans l'archive  $P_E$  si elle est non dominée dans  $P_I$  et si elle est non dominée dans  $P_E$ . Une fois, la solution insérée dans l'archive, on supprime tous les membres de l'archive qu'elle domine. Si l'ajout crée un dépassement de capacité de  $P_E$  membre de l'archive ayant le paramètre `squeeze_factor` le plus élevé est supprimé. Le paramètre `squeeze_factor` est égal au nombre d'individus qui appartiennent au même hypercube. Il est utilisé comme fitness des individus qui appartiennent à cette zone. Ce paramètre est utilisé pour la sélection ainsi que pour la mise à jour de l'archive alors que dans PAES, la mesure d'encombrement n'est utilisée que pour la mise à jour de l'archive.

Les critiques faites sur la méthode de crowding de PAES peuvent être réitérées pour PESA. La différence essentielle par rapport à PAES est que la sélection est basée sur la mesure d'encombrement de l'espace des objectifs. Si cela permet une bonne répartition des individus dans l'espace d'état, en contre-partie cela augmente la dépendance de l'efficacité de la méthode par rapport au facteur de discrétisation de l'espace.

#### 5.2.4. Region Based Selection (PESA II)

PESA II est une nouvelle technique de sélection basée sur l'utilisation d'hypercubes dans l'espace des objectifs, Corne (2001). Au lieu d'effectuer une sélection en fonction de la fitness des

individus comme dans PESA, cette méthode effectue une sélection par rapport aux hypercubes occupés par au moins un individu. Après avoir sélectionné l'hypercube, on choisit aléatoirement l'individu dans l'hypercube. Cette méthode se montre plus efficace à répartir les solutions sur la frontière de Pareto. Cela est dû à sa capacité de choisir avec une plus grande probabilité que le tournoi classique, des individus situés dans des zones désertiques, Figure (3.24). Par exemple dans la figure ci-dessous, les 10 points sont répartis dans 6 cubes. Si l'on considère un tournoi binaire alors la probabilité de sélectionner la solution A est : dans PESA  $1 - \left(\frac{9}{10}\right)^2 = 0.19$ , et dans PESA II :  $1 - \left(\frac{5}{6}\right)^2 = 0.31$ , [Berro, 2001].

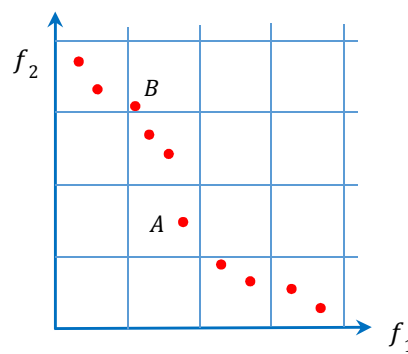


Figure (3.25) : Exemple de sélection par tournoi hypercube

PESA II a permis de faire évoluer positivement la sélection de manière à privilégier les zones de l'espace les moins encombrées. Mais, même si cette technique de crowding basée sur un découpage de l'espace est très supérieure en temps de calcul au sharing, elle possède ses propres difficultés. Dans la figure ci-dessus, le point B a la même probabilité de sélection que A, alors que B n'est pas situé dans une zone désertique Si l'on regarde la frontière de Pareto dans son ensemble. On peut aussi remarquer que si le découpage avait été de 4 cubes au lieu de 16 alors la probabilité de sélectionner la solution A serait passée de 0,31 à 0,89. Cela montre la très forte influence de la discrétisation de l'espace sur cette méthode, [Berro, 2001].

### 5.2.5. Non dominate Sorting Genetic Algorithm II (NSGA-II)

Pour la résolution des problèmes d'optimisation multi-objectif, de nombreux algorithmes génétiques ont été Développés. [Srinivas et al. 1994] ont proposé une autre variante de l'approche de Goldberg appelée -Non dominated Sorting Genetic Algorithm- (NSGA). La NSGA a connu un succès pendant plusieurs années [Weile et al. 1996], bien que plusieurs études comparatives de l'époque [Van Veldhuizen, 1999] ont montrées que NSGA a été devancé par MOGA et NPGA. NSGA est également un algorithme inefficace en raison de la façon dont il classe les individus.

En proposant le NSGA II, les chercheurs, [Deb et al. 2002] ont tenté de résoudre toutes les critiques faites sur NSGA, tel que, non-élitisme, complexité de calcul et utilisation de sharing qui implique le réglage d'un ou plusieurs paramètres. Dans cet algorithme, à chaque génération  $t$  une population de parents ( $P_t$ ) de taille  $N$  et une population d'enfants ( $Q_t$ ) de même taille sont assemblées pour former une population ( $R_t$ ) de taille  $2N$ . Cet assemblage permet d'assurer l'élitisme. La population ( $R_t$ ) est ensuite répartie en plusieurs fronts ( $F_1, F_2, \dots$ ) par une procédure de tri, plus rapide que celle proposée dans la première version de NSGA. Une nouvelle population parent ( $P_{t+1}$ ) est formée en ajoutant les fronts au complet (premier front  $F_1$ , second front  $F_2, \dots$  etc) tant que ceux-ci ne dépassent pas  $N$ . Si le nombre d'individus présents dans ( $P_{t+1}$ ) est inférieur à  $N$ , une procédure de crowding est appliquée. Le principe de fonctionnement de l'algorithme NSGA-II est résumé par les étapes suivantes et illustré par la Figure (3.25), Algorithme (3.8).

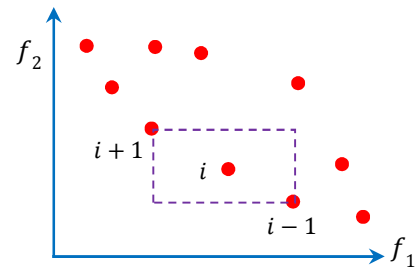
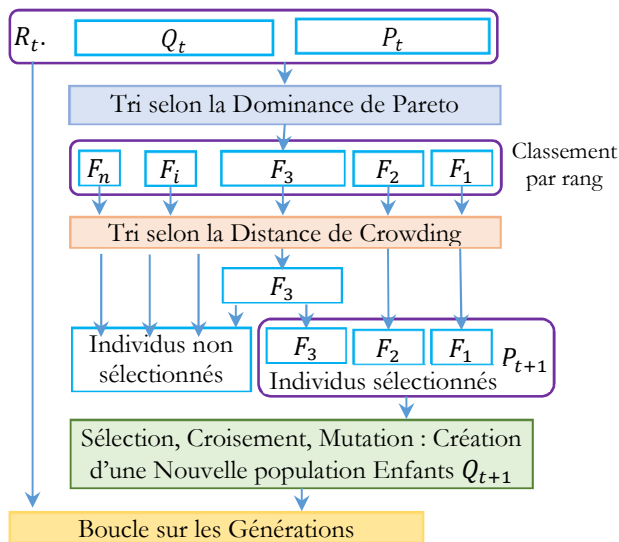


Figure (3.27) : Calcul d'un représentant (Crowding)

Figure (3.26) : Schéma de fonctionnement de NSGA-II

Algorithme NSGA-II
Initialiser les populations $P_0$ et $Q_0$ de taille $N$ <b>Tant que</b> critère d'arrêt non rencontré faire Création de $R_t = P_t \cup Q_t$ Calcul des différents fronts $F_i$ de la population $R_i$ par un algorithme de Ranking, Mettre $P_{t+1} = \phi$ et $i = 0$ , <b>Tant que</b> $ P_{t+1}  +  F_i  < N$ faire $P_{t+1} = P_t \cup F_i$ . $i = i + 1$ <b>Fin Tant Que</b> Inclure dans $P_{t+1}$ les $(N -  P_{t+1} )$ individus de $F_i$ les mieux répartis au sens de la distance de crowding Sélection dans $P_{t+1}$ et création de $Q_{t+1}$ par application des opérateurs de croisement et mutation <b>Fin Tant Que</b>

Algorithme (3.8) : Non dominate Sorting Genetic Algorithm II

Le calcul de valeur d'adaptation pour NSGA-II ne sert pas uniquement pour la sélection des opérateurs de croisement et de mutation, mais intervient aussi dans la sélection des individus à inclure dans  $P_{t+1}$  (la population contenant les élites). C'est donc une phase importante pour laquelle les auteurs de NSGA-II ont développé une méthode particulière : la distance de crowding.

### 5.2.5.1. Calcul de la distance de crowding

La distance de crowding  $d_i$  d'un point particulier  $i$  se calcule en fonction du périmètre de l'hypercube ayant comme sommets les points les plus proches de  $i$  sur chaque objectif. Sur la Figure (3.26) est représenté l'hypercube en deux dimensions associées au point  $i$ . Un algorithme de calcul de la distance de crowding est détaillé dans [Deb, 2001]. Cet algorithme est de complexité  $O(MN \cdot \log(N))$ , où  $M$  est le nombre d'objectifs du problème et  $N$  le nombre d'individus à traiter. Une fois tous les  $d_i$  calculés, il ne reste plus qu'à les trier par ordre décroissant et à sélectionner les individus possédant la plus grande valeur de crowding.

Cette nouvelle version de NSGA a permis de réduire la complexité de l'algorithme à  $O(k \cdot N^2)$ , de créer une méthode plus élitiste et de supprimer les paramètres de sharing.

## 5.3. Comparaison des approches multicritères

[Zitzler et al. 2001] comparent les approches d'optimisation multiobjectif et mettent en avant l'influence de l'introduction de l'élitisme sur la performance. Cependant la comparaison de la performance des méthodes élitistes dépendent des problèmes étudiés. Elle dépend de la méthode de comparaison de performance et la définition des paramètres pour chaque approche. [Roudenko, 2004] présente dans les comparaisons d'approches d'optimisations multicritères traitées dans la littérature.

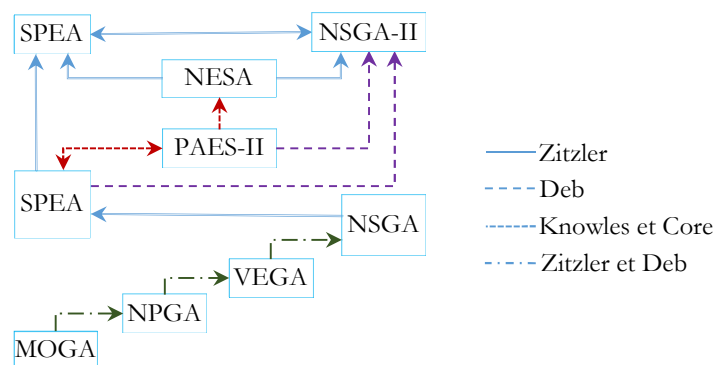


Figure (3.28) : Comparaison des performances des approches d'optimisation multicritère, [Roudenko, 2004].

Sur la base des développements présentés dans ce chapitre, nous avons établi une comparaison entre les différentes méthodes d'optimisation multicritère. Cette comparaison s'appuie sur des critères liés au principe de calcul, la performance de ces algorithmes de leurs avantages et inconvénients.

## 6. Problème de complexité

Après avoir abordé les différentes approches de résolution des problèmes d'optimisation multicritères, nous introduisons quelques notions sur la complexité de ces problèmes. La résolution de problèmes d'optimisation consiste à trouver, parmi un nombre de solutions potentielles très grand (mais fini), celle qui est optimale au sens d'un certain critère. La résolution de tels problèmes est capitale car ils apparaissent très souvent dans des problématiques industrielles diverses (logistiques, planification, ordonnancement de tâches, ...). On pourrait penser de prime abord, puisque, le nombre de solutions d'une instance d'un problème étant fini, alors, la puissance des ordinateurs permet de résoudre ce problème efficacement. Cette idée se heurte cependant à l'application pratique puisque le temps de calcul requis pour cet examen exhaustif est généralement extrêmement long. Le problème sera alors, de chercher des algorithmes de résolution efficaces.

Parmi les objectifs de la théorie de complexité est de classer les problèmes en fonction des ressources (temps de calcul, espace mémoire, etc...) nécessaires à leur résolution algorithmique. Ceci a montré qu'il existe des problèmes qui ont une solution calculable mais dont toute réalisation effective sur une machine est pratiquement inutilisable parce que le temps de calcul ou l'espace mémoire nécessaire sont trop importants. L'enjeu est donc de discerner entre les problèmes qui ont une solution réalisable et ceux qui, quelles que soient les améliorations futures des machines, ne peuvent intrinsèquement avoir une telle solution.

Généralement, le temps d'exécution est le facteur majeur qui détermine l'efficacité d'un algorithme, alors la complexité en temps d'un algorithme est le nombre d'instructions nécessaires (affectation, comparaison, opérations algébriques, lecture et écriture, etc.) que comprend cet algorithme pour la résolution d'un problème quelconque.

### 6.1. Complexité algorithmique

On appelle complexité en temps d'un algorithme le nombre d'instructions élémentaires mises en œuvre dans cet algorithme afin de résoudre un problème donné. Une instruction élémentaire sera une affectation, une comparaison, une opération algébrique, la lecture et l'écriture etc. Mais comme le décompte précis de toutes les instructions d'un programme risque d'être assez pénible, et qu'entre deux exécutions du même algorithme avec un jeu de paramètres différent, le nombre d'instructions exécutées peut changer, on se contentera, en général, d'apprécier un ordre de grandeur de ce nombre d'instructions. C'est ce qu'on désigne sous le nom de complexité de l'algorithme. Pour mesurer la complexité temporelle d'un algorithme, on s'intéresse plutôt aux opérations les plus coûteuses : Racine carrée, Log, Exp, Addition réelle, les comparaisons dans le cas des tris, etc.

## 6.2. Algorithmes polynomiaux et Algorithmes exponentiels

Un algorithme polynomial est un algorithme dont la complexité est  $O(p(n))$  où  $p$  est une fonction polynomiale et  $n$  dénote la longueur de données. Tout algorithme dont la complexité ne peut être bornée par un tel polynôme d'ordre  $n$ , est un algorithme exponentiel (bien que cette définition inclue certaines complexités non-polynomiales comme  $n$ , qui n'est pas considéré comme fonction exponentielle).

## 6.3. Les différentes classes de complexité

### 6.3.1. La réduction de Turing

On dit qu'un problème de recherche  $P_1$  se réduit polynomialement à un problème de recherche  $P_2$  par la réduction de Turing, s'il existe un algorithme  $A_1$  pour résoudre  $P_1$  en utilisant comme sous-programme l'algorithme  $A_2$  résolvant  $P_2$ , de telle sorte que la complexité  $A_1$  est polynomiale, quand on évalue chaque appel de  $A_2$  par une constante.

### 6.3.2. La classe P

La classe P contient tous les problèmes relativement faciles c'est à dire ceux pour lesquels on connaît des algorithmes efficaces. Plus formellement, ce sont les problèmes pour lesquels on peut construire une machine déterministe (de Turing) dont le temps d'exécution est de complexité polynomiale.

### 6.3.3. La classe NP

La classe NP contient les problèmes de décision qui peuvent être décidés sur une machine de Turing non déterministe en temps polynomial. C'est la classe des problèmes qui admettent un algorithme polynomial capable de tester la validité d'une solution du problème. Intuitivement, les problèmes de cette classe sont les problèmes qui peuvent être résolus en énumérant l'ensemble de solutions possibles et en les testant à l'aide d'un algorithme polynomial.

#### 6.3.3.1. La classe NP-complets

Parmi l'ensemble des problèmes appartenant à NP, il en existe un sous-ensemble qui contient les problèmes les plus difficiles : on les appelle les problèmes NP-complets. Un problème NP-complets possède la propriété que tout problème dans NP peut être transformé (réduit) en celui-ci en temps polynomial. C'est à dire qu'un problème est NP-complets quand tous les problèmes appartenant à NP lui sont réductibles. Si on trouve un algorithme polynomial pour un problème NP-complets, on trouve alors automatiquement une résolution polynomiale de tous les problèmes de la classe NP.

### 6.3.3.2. La classe NP-difficiles

Un problème est NP-difficile s'il est plus difficile qu'un problème NP-complet, c'est à dire s'il existe un problème NP-complet se réduisant à ce problème par la réduction de Turing.

Ceci explique pourquoi, lors de l'étude d'un nouveau problème, on commence par chercher à classer ce problème. Si l'on parvient à montrer qu'il est polynomial, le problème sera résolu. Si par contre, on parvient à montrer qu'il est NP-complet, la recherche d'un algorithme exact pour résoudre un tel problème ne sera pas de première priorité, et il sera approprié de se concentrer sur des méthodes heuristiques que la plupart des spécialistes de l'optimisation combinatoire ont orienté leurs recherches pour les développer. Une méthode heuristique est souvent définie comme une procédure exploitant au mieux la structure du problème considéré, dans le but de trouver une solution de qualité raisonnable en un temps de calcul aussi faible que possible.

## Conclusion

Dans la plupart des cas réels, on est confronté aux problèmes d'optimisation multiobjectifs, caractérisés par la présence simultanée de plusieurs objectifs, souvent contradictoires. En effet, les approches de résolution multicritère consistaient pendant longtemps, à la transformation des problèmes d'optimisation multicritères en problèmes monocritère. Plusieurs approches ont été développées dans ce contexte. Ces derniers nécessitent en réalité, des connaissances a priori sur le problème à traiter. En outre, elles modifient la structure du problème combinatoire, qui perd ainsi, ses éventuelles propriétés remarquables.

L'optimisation du problème monocritère peut garantir l'optimalité de la solution trouvée, mais n'en trouve qu'une seule, ce qui limite le choix des décideurs. Or, dans les situations réelles, les décideurs ont généralement besoin de plusieurs alternatives. La connaissance de plusieurs solutions optimales est utile aussi pour une utilisation future, particulièrement quand la situation change et qu'une nouvelle solution est requise. Le décideur peut ainsi choisir une solution suivant la situation courante. Ces approches sont sensibles au paysage de la frontière Pareto (convexité, discontinuité, etc.). Depuis quelques années, les approches Pareto ont été utilisées pour neutraliser les inconvénients des approches précédentes. Les approches Pareto présentent l'avantage de traiter les problèmes multicritères sans transformation, sans favoriser un objectif par rapport à un autre et peuvent générer des solutions supportées et non supportées en un passage unique de l'algorithme. Elles sont basées sur la notion de dominance définie comme une relation d'ordre partiel entre les solutions du problème.

Le choix d'une méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. Si le problème est de petite taille, alors un algorithme exact peut être utilisé pour trouver la

solution optimale. Malheureusement, ces algorithmes par nature énumératifs, souffrent de l'explosion combinatoire et ne peuvent s'appliquer à des problèmes de grandes tailles. Dans ce cas, il est nécessaire de faire appel à des métaheuristiques permettant de trouver de bonnes solutions approchées.



---

**CHAPITRE 4.**

**OPTIMISATION MULTICRITÈRE**

**D'UN JSSP INTÉGRÉE DE LA**

**MAINTENANCE ET DE LA**

**PRODUCTION**

---

# Chapitre 4.

## Optimisation multicritères d'un JSSP intégrée des activités de la maintenance et de la production

---

***Résumé :** Dans ce chapitre nous étudions le problème d'optimisation multicritères d'ordonnancement intégrant les activités de production et de maintenance dans un atelier de type job shop. Les critères d'optimisation considérés sont le Makespan, le coût de production et le coût de maintenance.*

*Nous présentons en premier lieu une description détaillée du problème considéré suivie d'une formulation mathématique. En second lieu, pour la résolution du problème, nous développons l'algorithme non-dominated sorting genetic algorithm (NSGA-II), ainsi que le codage et les opérateurs appropriés. Les résultats d'expériences effectuées sur les instances générées selon un schéma particulier, permettent de montrer l'efficacité des approches proposées.*

---

## Introduction

Le problème ordonnancement job shop (JSSP) est depuis longtemps l'objet d'étude dans l'environnement académique et industriel. La littérature actuelle, compte plusieurs recherches qui offrent des états de l'art sur les JSSP. Ces études ont mis en valeur les principales techniques de résolution, [Jain et al. (1999), [Sha et al. 2010], [Voratas et al. 2011], [Manas et al. 2015]. De nombreux auteurs ont démontré que le JSP est la classe la plus difficile des problèmes d'optimisation combinatoire, entre autres [Garey et al. 1976] ont prouvé que les JSSP sont NP-difficile, raison pour laquelle nous ne pouvons pas trouver pour ce type de problèmes des solutions exactes dans un temps de calcul raisonnable.

Parmi les problèmes d'ordonnancement job shop, les JSSP mono-objectif sont les plus investigués. La plupart d'entre eux fixent comme objectif la minimisation du temps maximal de séjour des jobs dans l'atelier le Makespan ( $C_{max}$ ). Pour surmonter les limites des techniques d'optimisation exacte de nombreuses approches de résolution approchées (heuristiques) ont été développées. Ces approches comprennent le recuit simulé (SA) [Liang et al. 2005], [Kouider et al. 2007], [Gorine Ali, 2011], la recherche de tabou, [Sun et al. 1995], [Nowicki et al. 1996], [Pezzella et al. 2000], [Anthony 2006], les algorithmes génétiques (GA), [Rui, 2012], [Lei et al. 2014], [Jingyao et al. 2015], les colonies de fourmis, [Apinanthana et al. 2008], [Przemyslaw et al. 2013], [Boxuan et al. 2015] et essais particuliers, [Maroua et al. 2015], [Wannaporn et al. 2016].

Cependant, les systèmes de production dans le monde réel exigent la satisfaction simultanée de plusieurs objectifs, généralement partiellement antagonistes. En effet, divers versions d'algorithmes multicritères ou multicritère évolutionnistes ont été développé, (voir chapitre 3). Étant donné l'importance de la production et de la maintenance dans les entreprises manufacturières, ces algorithmes ont trouvés leur application dans la résolution du problème d'ordonnancement intégrant les activités de maintenance et de production.

Le problème d'ordonnancement job-shop multicritère (MCJSSP) est l'un des problèmes d'ordonnancement qui présente un intérêt théorique et pratique toujours renouvelé pour les chercheurs surtout avec la prise en compte de la disponibilité des ressources, justifiée par de multiples raisons pratiques [Youssef, 2003], [Fitouhi, 2011]. Suivant la période de disponibilité, quatre principaux groupes de MCJSSP sont à distinguer : à intervalle de disponibilité fixe, à plusieurs périodes de disponibilité fixes, à périodes de disponibilités flexibles et à plusieurs périodes de disponibilité flexibles [Azem et al. 2011]. D'après la littérature, la plupart des MCJSSP sont transformés par agrégation à un JSSP monocritère. Ces problèmes peuvent être résolus par l'un des algorithmes d'optimisation monocritère.

Cependant, les algorithmes de résolution utilisés sont généralement dominés et sous-estimés par la nouvelle génération des algorithmes appelés algorithmes évolutionnaires multi-objectifs à base de Pareto (MOEA) [Deb et al. 2005]. Parmi ces algorithmes l'algorithme NSGA (non-dominated sorting genetic algorithm) proposé par [Srinivas et al. 1995] est le plus reconnu. Toutefois, dans le but d'améliorer d'avantage les performances de NSGA, [Deb et al. 2002] et [Deb et al. 2000], ont étendu NSGA à un algorithme plus efficace appelé NSGAI. Bien que cette version de l'algorithme ait la même appellation que la précédente, la différence reste fondamentale. [Deb et al. 2002] ont également prouvé que la complexité des NSGAI est de type ( $O(MN^2)$ ) est inférieure à celle de NSGA.

Dans le but de se rapprocher davantage de la réalité industrielle des entreprises, nous considérons dans cette thèse le JSSP-multicritères, où non seulement les contraintes temporelles et de disjonction, mais aussi, ceux de la disponibilité des ressources. Les périodes de disponibilité sont supposées flexibles, dans le sens qu'elle varie en fonction de la charge des ressources et que dans tous les cas de figure, les périodes d'indisponibilité sont calculées aprioris, en exploitant l'historique des machines et supposés constant le long de l'horizon de l'ordonnancement. La tâche de maintenance est considérée elle aussi flexible dans le sens qu'elle peut glisser dans une fenêtre de temps. Pour garantir des produits à moindre coût et des clients servis en Just-à-temps, trois critères sont adoptés : la date d'achèvement du job le plus tardif, le coût de production et le coût de maintenance.

Vu la complexité conjuguée de notre problème, l'algorithme NSGA-II « Non-dominated Sorting Genetic Algorithm – II » est jugés performant et s'adaptent bien au traitement des problèmes d'optimisation combinatoire multi objectifs. En plus, la littérature témoigne que cette métaheuristique connaît une grande efficacité pour résoudre les problèmes de type MCJSSP.

Ce chapitre est organisé en cinq sections. Dans la première section, une description du problème considéré sera décrite. Dans la deuxième, la formulation mathématique des trois critères et les différentes contraintes du MCJSSP seront détaillés. La troisième section est réservée à l'application de NSGA-II au MCJSSP, où la description de l'algorithme NSGA-II et la proposition des opérateurs génétiques spécifique au problème ont été présenté. Ensuite dans la quatrième section les résultats d'expérimentations seront discutés où un exemple illustratif des concepts présentés à l'aide de l'instance ft06. En fin, une conclusion sera présentée dans la dernière section.

## 1. Description du problème d'ordonnancement

Le problème d'ordonnancement job shop multicritères noté MCJSSP avec contraintes de disponibilité des ressources peut être défini par un ensemble de  $n > 1$  jobs (lots) différents  $J = \{J_i\}_{i=1}^{i=n}$ , à réaliser sur un ensemble de  $m > 1$  machines  $M = \{M_k\}_{k=1}^{k=m}$  machines. Ces machines sont des ressources critiques dans le sens où une machine ne peut être affectée à deux jobs simultanément. Chaque Job  $J_i \in J$  est décrit par une gamme opératoire appropriée. Une gamme définit l'ordre, la durée et la séquence des machines où sont traitées les opérations d'un job. Chaque job  $J_i \in J$  est composé d'une séquence linéaire de  $n_i$  opérations  $J_i = \{O_{ij}\}_{j=1}^{j=n_i}$  avec  $1 \leq n_i \leq m$  et limité par une échéance  $d_i$  (due date) avant laquelle il doit obligatoirement s'achever. Les  $J_i$  sont soumises à des pénalités d'avance et de retard relativement à  $d_i$ . Chaque machine peut effectuer au maximum une seule opération à la fois et chaque opération  $O_{ij}$  requiert une machine unique pour sa réalisation, et ce pendant  $p_{ij}$  unités de temps sa date de début est  $s_{ij}$  et sa date de fin est  $c_{ij}$ . Cette opération doit être réalisée sans préemption.

L'ensemble des opérations à ordonnancer est noté par  $\Phi$ , équation (4.1) :

$$\Phi = \bigcup_{1 \leq i \leq n} J_i \quad \text{où} \quad \Phi = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n_i}} O_{ij} \quad (4.1)$$

L'ensemble des opérations affectées à une même ressource  $M_k$  est noté par  $\Phi_k$  équation (4.2)

$$\Phi_k = \bigcup_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n_i}} (O_{ij} \times \alpha_{ijk}) \quad \text{avec} \quad \alpha_{ijk} = \{0,1\} \quad (4.2)$$

$$\text{tel que: } \alpha_{ijk} = \begin{cases} 1: & \text{si } O_{ij} \text{ est affectée à } M_k \\ 0: & \text{si non} \end{cases}$$

Le nombre d'opérations de production à traiter sur la machine  $M_k$  est donc  $e_k = |\Phi_k|$ . Les opérations de  $\Phi_k$  doivent donc être ordonnées suivant une heuristique appropriée, on dit qu'il y a disjonction entre elles car les intervalles de temps pendant lesquels la machine est occupée sont disjoints.

Pour des exigences technologiques, chacune des machines est soumise périodiquement à des périodes d'indisponibilité  $\tau_k$  pour l'intervention préventive systématique de durée  $tp_{kh}$ . La durée moyenne d'intervention corrective est  $tr_{kh}$ . Les machines subissent des interventions pour maintenance. On suppose qu'il y a  $m_k$  périodes d'indisponibilité sur chaque machine  $M_k$ . Ces périodes d'indisponibilité définissent par l'ensemble des opérations de maintenance

$\{Om_{kh}\}_{h=1}^{h=m_k}$  affecté à la machine  $M_k$ . Nous supposons qu'elles ne sont pas trop proches. La date de début  $sm_{kh}$  de la période d'indisponibilité  $Om_{kh}$  de durée  $pm_{kh}$  sont calculés aprioris, en exploitant l'historique des machines et supposés constante le long de l'horizon de l'ordonnancement. Dans cette thèse nous supposons aussi que,  $Om_{kh}$  est flexible, notion introduite dans [Aggoune, 2004] et [Azem, 2010]. C'est-à-dire ont associé à chaque  $sm_{kh}$  une fenêtre de temps  $[esm_{kh}, lsm_{kh}]$ , où  $esm_{kh}$  et  $lsm_{kh}$  sont respectivement les dates de début au plus tôt et au plus tard, entre lesquelles  $sm_{kh}$  peut glisser. L'idée est de créer un temps libre sur la machine pour exécuter une opération soit avant la période d'indisponibilité soit après ce qui permet à l'opération de s'achever plus tôt. La préemption des opérations de production par ceux de la maintenance est non-autorisée.

## 2. Formulation du MCJSSP

Afin de décrire le problème Job-Shop multicritères de façon univoque, nous décrivons dans un premier temps à l'aide du formalisme mathématique.

Le problème d'ordonnancement Job-Shop multicritères présente la difficulté de l'affectation des opérations de chaque produit aux machines requises, et celle relative à l'ordonnancement de l'ensemble des jobs dans un ordre minimisant un certain nombre de critères.

Nos travaux s'alignent avec les concepts de la démarche de production Just à temps (JAT), dont la principale motivation est de fabriquer plusieurs produits en lots de volume réduit et d'éliminer les tout formes de gaspillage à tous les niveaux. Dans ce contexte vient le choix, du MCJSSP, avec lequel, nous recherchons un ordonnancement réalisable qui minimise simultanément trois critères. Le premier est le Makespan, le second est les coûts de production et le troisième les coûts de maintenance en introduisant pour les deux derniers la notion de pénalité Avance/Retard. Ces critères sont optimisés simultanément et sans agrégation en utilisant un algorithme d'optimisation multicritère. Ceci en satisfaisant un ensemble de contraintes, d'enchaînement, de disjonction et d'indisponibilité des ressources pour maintenance.

### 2.1. Makespan ( $\min_{1 \leq i \leq n} (C_{\max i})$ )

Le Makespan est un critère d'optimisation, représente la date de fin d'exécution de la dernière opération qui quitte l'atelier. Il est particulièrement intéressant puisque, il permet de déterminer les séquences critiques, c'est-à-dire les séquences sur lesquelles tout retard a des conséquences sur le temps de réalisation des jobs. Sa minimisation implique la minimisation de la durée de séjours des produits dans l'atelier, des encours ainsi que le temps non productif des machines.

Makespan est la date de fin du dernier job qui sort de l'atelier de production. Il est défini par l'équation (4.3)

$$f(1) = \min_{1 \leq i \leq n} \left( \max_{1 \leq j \leq n_i} (c_{ij}) \right) \quad (4.3)$$

## 2.2. Coût de production

Les coûts de la production ( $C_{prod}$ ) sont définis comme étant la somme du prix d'achat des matières premières et l'ensemble des charges dès l'acquisitions jusqu'à la mise sur stock des produits finis apte à la consommation. Autrement dit c'est la somme du coût d'acquisition des matières premières, charges de stockage, charges d'utilisation des lignes de production (avec charges de préparation et charges de transport entre les différents postes de travail), charge de pénalités.

**Coût de production = coût de matière première + coût de stockage + coût de fabrication + coût de pénalités.**

Les coûts de production sont calculés par produit (job) et représentés par l'équation (4.4)

$$C_{PT} = \sum_{1 \leq i \leq n} C_{P_i} = \sum_{1 \leq i \leq n} C_{Mp_i} + C_{S_i} + (C_{F_i} + C_{A_i} + C_{R_i}) \quad (4.4)$$

### 2.2.1. Coût de matière première

Le coût de matière première est constitué du prix d'achat des matières premières nécessaire à la fabrication des produits finis. Si chaque job  $J_i$  requiert  $u_i$  matières premières, dont les quantités correspondantes  $q_{ij}$ , au prix unitaire  $PU_{ij}$ . Le coût de matière est défini par l'équation (4.5).

$$C_{Mp_i} = \sum_{1 \leq j \leq u_i} PU_{ij} \times q_{ij} \quad i \in \{1, 2, \dots, n\} \quad (4.5)$$

### 2.2.2. Coût de stockage

Le coût total de stockage de chaque produit ( $C_{S_i}$ ) est égal à la somme du coût de détention ( $C_{d_i}$ ), dit aussi coût de possession ( $C_{L_i}$ ) et du coût de lancement (dit aussi coût de passation), équation (4.6)

$$C_{S_i} = C_{d_i} + C_{L_i}$$

$$C_{S_i} = S_{M_i} \times \tau_{d_i} + N_{c_i} \times C_{K_i} \quad (4.6)$$

### 2.2.3. Coût de détention

Les différents éléments qui entrent dans le coût de stockage d'une entreprise sont classés en deux composantes. La première consiste en le coût de détention du stock, qui regroupe tous les éléments engendrés par la seule possession du stock, telle que la location des entrepôts où sont placées les matières premières, l'assurance des produits, l'amortissement des matières ainsi que les coûts des capitaux qui financent les stocks. Par conséquent, ce coût dépend des niveaux des stocks. Le coût de détention est le produit du Stock moyen de la période ( $S_{M_i}$ ) et le taux de détention ( $\tau_{d_i}$ ) équation (4.7).

$$C_{d_i} = S_{M_i} \times \tau_{d_i} \quad (4.7)$$

### 2.2.4. Coût de lancement

La deuxième composante du coût de stockage est le de lancement des commandes, ce dernier est engendré chaque fois que l'entreprise va passer une commande à ses fournisseurs.

En production, le coût de lancement se compose d'un coût administratif et d'un coût technique :

- Coût administratif : Estimation du coût d'élaboration et de gestion d'un Ordre de Fabrication (OF)
- Coût technique : dépenses liées au changement de fabrication : Temps machine (immobilisée pendant le changement) et Temps main d'œuvre (participant au changement). Il est à noter que, le temps de changement de fabrication se mesure de la dernière pièce d'une série jusqu'à la première pièce bonne de la série suivante.

En approvisionnement, on trouvera un découpage similaire :

- Coût de passation de commande : Estimation du coût d'élaboration et de gestion d'un ordre d'approvisionnement (OA - commande fournisseur).
- Coûts "physiques" fixes liés à la commande, par exemple, de transport et de réception.
- Le coût de lancement est le produit du Nombre des commandes de la période ( $N_{c_i}$ ) et le Coût de lancement d'une commande ( $C_{K_i}$ ) équation (4.8).

$$C_{L_i} = N_{c_i} \times C_{K_i} \quad (4.8)$$

### 2.2.5. Coût de fabrication ( $C_{F_i}$ )

Le coût de fabrication est composé du coût d'utilisation des moyens de production : amortissement des machines, énergie, main d'œuvre, etc., équation (4.9)



$$C_{Fi} = \sum_{1 \leq k \leq m} \sum_{1 \leq j \leq n_i} (C_{F_{u_{ik}}} \times p_{ij} \times \alpha_{ijk}) \quad (4.9)$$

### 2.2.6. Coût de pénalité

Dans un environnement économique caractérisé, d'une part, par des délais d'attente relativement plus restreints incitant à produire à temps, d'autre part, par la variabilité très importante de la demande pour satisfaire sa clientèle, et d'une très forte concurrence. De ce fait, il convient de développer des systèmes de production capables d'offrir aux clients une grande variété de produits, au bon moment, au moindre coût, et sans engendrer de stocks et des en-cours inutiles. Ainsi est né le concept de production juste à temps (JAT). Pour s'aligner avec cette démarche de production, les problèmes d'ordonnancement utilisent une fonction objectif liée à la minimisation simultanée des coûts engendrés par la production anticipée, ou tardive des produits. Deux critères sont ainsi considérés : un critère d'avance qui pénalise un produit exécuté tôt et un critère de retard qui pénalise un produit traité en retard.

Le coût de pénalité de chaque produit est la somme du coût de pénalité d'avance et du coût de pénalité de retard, par rapport à une date de fin souhaitée, équation (4.10).

Coût de pénalité =  $ap_i \times E_i + rp_i \times T_j$ .

$$= ap_i \times \max_{1 \leq i \leq n} (0, (d_i - c_{in_i})) + rp_i \times \max_{1 \leq j \leq n} (0, (c_{in_i} - d_j)) \quad (4.10).$$

Les poids  $ap_i$  et  $rp_i$  représentent l'importance du produit (job) au sein du système de production. Cette fonction pénalise tout produit se terminant avant ou après sa date d'échéance. En effet, la minimisation de l'avance permet de réduire les coûts générés par les stockages inutiles et/ou la détérioration des produits, tandis que la minimisation du retard a pour rôle de réduire les coûts engendrés par le non-respect de la date de livraison et/ou la perte des clients. Il s'agit donc de trouver un compromis entre ces coûts. Ce critère est lié à la catégorie de critères d'optimisation irréguliers (Gupta et al. 2002).

La date d'échéance  $d_i$  d'un lot  $i$  est calculée par une formule équation (4.11), Cette formule permet à l'utilisateur de choisir une date d'échéance cohérente pour tous les lots.

$$d_i = \sum_{1 \leq j \leq n_i} \frac{\sum_{1 \leq k \leq m} p_{ij} \times \alpha_{ijk}}{m} \times cd \quad (4.11)$$

avec  $p_{ij} \times \alpha_{ijk}$  le temps de réalisation de l'opération  $O_{ij}$  sur la machine  $M_k$  et  $cd$  le coefficient pour la date d'échéance.

### 2.2.6.1. Coût de pénalité d'avance ( $C_{A_i}$ )

Un produit réalisé avant sa date de livraison ne risque pas seulement de générer des charges de stockage inutile, mais aussi de perte de valeur en particulier pour les produits périssables tels que les produits agroalimentaires, pharmaceutiques, etc.

Les coûts de pénalités d'avance augmentent en fonction des avances  $E_i$ . Figure (4, 1)

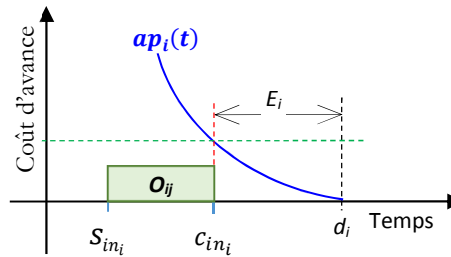


Figure (4, 1) : pénalités d'avance

Soit  $ap_i$  pénalité unitaire d'avance supposée linéaire.

Les coûts de pénalité d'avance sont calculés par l'équation (4.12)

$$C_{A_i} = ap_i \times E_i \quad (4.12)$$

Avec la fonction des avances  $E_i$  est défini par l'équation (4.13)

$$E_i = \max_{1 \leq i \leq n} (0, (d_i - c_{in_i})) \quad (4.13)$$

### 2.2.6.2. Coût de pénalité de retard ( $C_{R_i}$ )

Un produit réalisé après sa date de livraison génère des pénalités sous forme : perte de clients (manque à gagner), perte de la qualité de service clients, etc. Les coûts de pénalités de de retard augmentent en fonction des durés de retard  $T_i$ , Figure (4, 2).

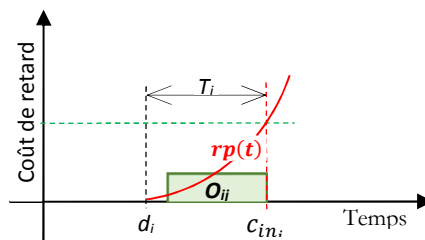


Figure (4, 2) : pénalités de retard

Soit  $rp_i$  pénalité unitaire de retard supposée linéaire.

Les coûts de pénalité de retard sont calculés par l'équation (4.14)

$$C_{R_i} = rp_i \times T_j \quad (4.14)$$

Avec la fonction des retards  $T_i$  est défini par l'équation (4.15)

$$T_i = \max_{1 \leq i \leq n} (0, (c_{in_i} - d_i)) \quad (4.15)$$

### 2.2.7. Le coût de production total $C_{PT}$

Le coût de production total représente l'ensemble des coûts, de matière première, de stockage, de fabrication et de pénalités, équation (4.16). Dans ce travail nous ne tenons pas compte des prix d'acquisition des matières premières. Car, bien quelle influe énormément sur le prix de revient des produits finis, mais elle reste une cause d'influence extrinsèque.

$$C_{PT} = \sum_{1 \leq i \leq n} C_{P_i} = \sum_{1 \leq i \leq n} C_{MP_i} + C_{S_i} + (C_{A_i} + C_{R_i} + C_{F_i}) \quad (4.16)$$

Alors, la deuxième fonction objectif est défini par l'équation (4.17)

$$f(2) = \sum_{1 \leq i \leq n} \left( C_{MP_i} + (S_{M_i} \times \tau_{d_i} + N_{c_i} \times C_{K_i}) + (ap_i \times E_i + rp_i \times T_i) \right. \\ \left. + \sum_{1 \leq k \leq m} \sum_{1 \leq j \leq n_i} (C_{Fu_{ik}} \times p_{ij} \times \alpha_{ijk}) \right) \quad (4.17)$$

### 2.3. Coût de maintenance

Par cette fonction objectif nous cherchons la minimisation des coûts de maintenance composé dans cette étude de : coût de pièce de rechange, coûts d'intervention préventive et coûts de pénalité d'avance de de retard, et de maximiser en revanche la disponibilité des machines. Nous supposons que :

- Les coûts de la main d'œuvre de maintenance sont constants,
- la période de disponibilité optimale des machines calculée et supposée constante,
- l'adoption d'une politique de maintenance préventive, pour réduire le taux de défaillance des machines est justifié, avec un paramètre de forme ( $\beta > 1$ ),
- le coût horaire d'intervention est constant noté  $C_{I_k}$ ,

- Après chaque intervention, la machine est restaurée à ses performances optimales, alors, l'exploitation de la machine peut être modélisée comme un processus de renouvellement.
- la réparation est minime.

Les dates de début et de fin de la  $h^{\text{ème}}$  opération de maintenance préventive, noté respectivement  $sm_{kh}$  et  $cm_{kh}$  sont reliées par l'équation (4.18).

$$cm_{kh} = sm_{kh} + pm_{kh} \quad \forall k = \{1, 2, \dots, m\}, \quad \forall h = \{1, 2, \dots, m_k\} \quad (4.18)$$

Avec  $pm_{kh}$  la durée d'indisponibilité (de remise en bon état de la machine) est supposée constante et égale au temps moyen d'intervention.

$$pm_{kh} = tp_k \quad \forall k = \{1, 2, \dots, m\}, \quad \forall h = \{1, 2, \dots, m_k\}$$

La date d'échéance de l'intervention préventive systématique  $dm_{kh}$  défini pour chaque intervention, en fonction de la charge de la machine  $M_k$  dans la période  $h$ . Figure (4. 3).

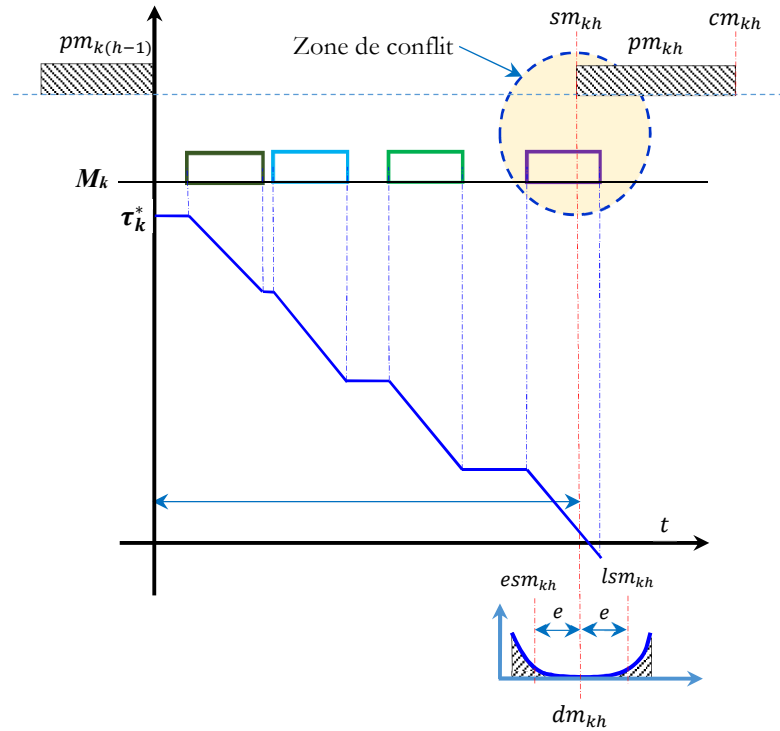


Figure (4.3) : dates d'intervention souhaitée et période d'intervention.

### 2.3.1. Calcul de la période de disponibilité opérationnelle :

En réalité chaque machine  $M_k$  de l'atelier est soumise à un programme d'intervention périodique de périodicité  $\tau_k$  unités de temps, pour maintenance préventive systématique et à des interventions probables (non programmées) pour maintenance corrective.

La question qui se pose est de savoir comment déterminer la périodicité des interventions de maintenance préventive systématiques (MPS). Traditionnellement, l'intervalle  $\tau_k$  est déterminé en fonction des expériences. Nous proposons dans cette thèse pour calculer la périodicité optimale  $\tau_k^*$  l'application de la loi de weibull, basée sur l'exploitation de l'historique de la machine en question.

La loi de weibull est Proposée par l'ingénieur et mathématicien suédois Ernst Hjalmar Waloddi Weibull (1887-1979). En effet, cette loi s'adapte bien à l'étude statistique des défaillances, en particulier dans la période de la jeunesse et de l'usure de la machine. La distribution de Weibull est une sorte de loi caméléon, très souple, qui, grâce à ses trois paramètres ( $\gamma, \alpha$  et  $\beta$ ), peut s'ajuster à un grand nombre de données statistiques. Elle peut suivre une distribution non symétrique, faire l'approximation de la loi normale, devenir une distribution exponentielle, etc. Elle est également utilisée pour décrire des phénomènes de fatigue, des durées de vie (roulements...), des probabilités de rupture sous charge, etc. En raison de sa flexibilité, la loi de Weibull est très utile pour modéliser la fiabilité et la disponibilité de divers équipements à partir d'un retour d'expérience. Elle est caractérisée par :

La densité de probabilité, équation (4.19).

$$f(t) = \frac{\beta}{\eta} \left( \frac{t - \gamma}{\eta} \right)^{\beta-1} \times \exp \left[ - \left( \frac{t - \gamma}{\eta} \right)^\beta \right] \quad (4.19)$$

La fiabilité  $R(t)$  et la fonction de répartition  $F(t)$ , équation (4.20).

$$R(t) = \exp \left[ - \left( \frac{t - \gamma}{\eta} \right)^\beta \right] \quad \text{et} \quad F(t) = (1 - R(t)) \quad (4.20)$$

Taux de défaillance  $\lambda(t)$  :

$\lambda(t)$  est fonction de probabilité de défaillance de la machine, équation (4.21).

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{\beta}{\eta} \left( \frac{t - \gamma}{\eta} \right)^{\beta-1} \quad (4.21)$$

Avec  $\beta_k$  : Paramètre de forme de  $M_k$ ,  $\eta_k$  : facteur d'échelle de  $M_k$  et  $\gamma_k$ : paramètre de position de  $M_k$ .

$\tau_k$  désigne la variable aléatoire qui, à chaque matériel, associe son temps de bon fonctionnement (Time Between Failures) ou sa durée de vie avant une défaillance. Son espérance mathématique et la moyenne des temps de bon fonctionnement ou MTBF. Pour simplifier, nous choisissons comme origine des temps l'instant ( $t = 0$ ) où le dispositif choisi est mis en marche, soit

pour la première fois, soit après une réparation qui l'a remis à neuf. Alors  $\tau_k$  mesure ainsi l'instant où apparaît la première défaillance d'un dispositif pris au hasard dans la population considérée, à partir de l'instant ( $t = 0$ ).

Soit  $N_k(t)$  désigne le nombre de défaillances qui se produisent sur la machine  $M_k$  au cours d'un cycle donné. La valeur attendue de  $N_k(t)$  est défini par l'équation (4.22).

$$N_k(t) = \left(\frac{t}{\eta_k}\right)^{\beta_k} \quad (4.22)$$

La «moyenne» du cycle se compose des unités de temps de fonctionnement,  $m(t)$  réparations de durée moyenne  $tr_k$ , et un intervalle de maintenance préventive de durée  $tp_k$ . L'état d'équilibre de disponibilité résultant de la machine est une fonction de l'intervalle de PMS et donnée par l'équation (4.23).

$$A_k(t) = \frac{t}{t + N_k(t) \times tr_k + tp_k} \quad (4.23)$$

Le temps de bon fonctionnement entre deux interventions préventives successives sur chaque machine  $M_k$  est donné par l'équation (4.24).

$$\tau_k^* = \eta_k \left[ \frac{tp_k}{(\beta_k - 1)tr_k} \right]^{\frac{1}{\beta_k}} \quad (4.24)$$

### 2.3.2. Coût d'avance et de retard des interventions

Pour traduire la volonté de maintenir les machines de production selon la politique du Just-à-temps, les programmes de planification des interventions pour maintenance préventive systématique utilisent une fonction objectif liée à la minimisation simultanée des coûts engendrés par les opérations anticipées, ou tardives des interventions. Deux critères sont ainsi considérés : un critère d'avance qui pénalise une intervention exécutée trop tôt et un critère de retard qui pénalise une intervention traitée en retard. Pour chaque intervention  $h$  sur une machine  $M_k$  ayant une date de début souhaitée  $dm_{kh}$  et une date de fin réelle,  $cm_{kh}$  nous définissons les coûts d'avance et de retard de la manière suivante :

#### 2.3.2.1. Coûts d'avance des interventions

L'avance de l'intervention sur la machine  $M_k$  est la différence entre la date d'intervention souhaitée et la date de de début effective de la  $h^{ième}$  opération, équation (4.25).

$$Em_{kh} = \max[0; (dm_{kh} - sm_{kh})] \quad (4.25)$$

L'avance cumulée sur chaque machine  $M_k$  est donnée par l'équation (4.26).

$$EM_k = \sum_{1 \leq h \leq m_k} Em_{kh} \quad (4.26)$$

Les coûts de pénalité d'avance traduisent les coûts de changement précoce des pièces encore non consommées. Ces coûts sont proportionnels à la durée cumulée des avances des interventions  $EM_k$  sur chaque machine  $M_k$ . Sachant que,  $am_k$  est la pénalité unitaire d'avance (coût du résidu de la pièce de rechange). Elle est supposée constante.

Le coût d'avance total est défini donc par l'équation (4.27)

$$CA_k = am_k \times EM_k \quad (4.27)$$

### 2.3.2.2. Coût de retard des interventions

Le retard de l'intervention sur la machine  $M_k$  est la différence entre la date de début effective de la  $h^{ième}$  intervention et la date d'intervention souhaitée correspondante, équation (4.28).

$$Tm_{kh} = \max[0; (sm_{kh} - dm_{kh})] \quad (4.28)$$

Le retard cumulé des interventions sur chaque machine  $M_k$  sera calculé par l'équation (4.29).

$$TM_k = \sum_{1 \leq h \leq m_k} Tm_{kh} \quad (4.29)$$

Le coût de pénalité de retard traduit le coût du risque de perte de production suite à des défaillances très probables à cause de non-respect du programme de maintenance préventive. Il est proportionnel à la durée cumulée des retards  $TM_k$ , équation (4.30).

$$CR_k = rm_k \times TM_k \quad (4.30)$$

$rm_k$ : Pénalité unitaire de retard supposée constante. Elle est affectée aux machines  $M_k$ .

### 2.3.2.3. Coût d'avance et de retard des interventions

Le coût de pénalité de la machine  $M_k$  est donc la somme des coûts de pénalité d'avance et des coûts de pénalité de retard, équation (4.31)

$$CP_k = am_k \times EM_k + rm_k \times TM_k$$

$$CP_k = am_k \times \sum_{1 \leq h \leq m_k} Em_{kh} + rm_k \times \sum_{1 \leq h \leq m_k} Tm_{kh} \quad (4.31)$$

Le coût d'intervention total sur l'ensemble des machines est défini par l'équation (4.32).

$$\sum_{1 \leq k \leq m} CP_k = \sum_{1 \leq k \leq m} \left( am_k \times \sum_{1 \leq h \leq m_k} Em_{kh} + rm_h \times \sum_{1 \leq h \leq m_k} Tm_{kh} \right) \quad (4.32)$$

### 2.3.3. Coût d'interventions de maintenance

Les coûts d'intervention ( $C_{int}$ ) représentent l'ensemble des charges : coût des pièces de rechange, charge du personnel d'intervention, charges d'administration en relation, charge d'utilisation de l'outillage, etc. ces coûts sont proportionnels à la durée d'intervention  $Om_{kh}$ , équation (4.33). Dans ce qui suit, nous définissons les paramètres de calcul des coûts d'intervention

$\mu_{kh} \in \{0,1\}$  : Coefficient d'intervention sur la machine  $M_k$ .

$$\mu_{kh} = \begin{cases} 1 & \text{Si la } h^{\text{ième}} \text{ intervention est exécutée sur la machine } M_k \\ 0 & \text{Si non} \end{cases}$$

$\sigma_k$  : Coût unitaire d'intervention (suppose constant) sur la machine  $M_k$ .

$pm_{hk}$  : Durée de la  $h^{\text{ième}}$  d'intervention sur la machine  $M_k$

$$C_{int} = \sum_{1 \leq k \leq m} \sum_{1 \leq h \leq m_k} (\mu_{kh} \times \sigma_k \times pm_{kh}) \quad (4.33)$$

### 2.3.4. Coût total de maintenance

Le Coût total de maintenance qui représente le troisième critère à minimiser est la somme des coûts d'avance et de retard et les coûts d'intervention, équation (4.34).

$$f(3) = \min \left[ \sum_{1 \leq k \leq m} \left( am_k \times \sum_{1 \leq h \leq m_k} Em_{kh} + rm_h \times \sum_{1 \leq h \leq m_k} Tm_{kh} \right) + \sum_{1 \leq k \leq m} \sum_{1 \leq h \leq m_k} (\mu_{kh} \times \sigma_k \times pm_{kh}) \right] \quad (4.34)$$

## 2.4. Contraintes

En réalité plusieurs contraintes sont à prendre en considération lors de l'optimisation des critères.

### 2.4.1. Contraintes de positivité

Les contraintes de positivité assurent que les variables de décisions : dates de début, les dates de fin et les durées d'exécution des opérations de production et de maintenance à ordonnancer soient toujours positives ou nulles, les inégalités (4.35)



$$\begin{aligned}
 s_{ij} &\geq 0 \\
 c_{ij} &\geq 0 && \forall i \in \{1,2, \dots, n\} \\
 p_{ij} &\geq 0 && \forall j \in \{1,2, \dots, n_i\} \\
 sm_{kh} &\geq 0 && \forall k \in \{1,2, \dots, m\} \\
 cm_{kh} &\geq 0 && \forall h \in \{1,2, \dots, m_k\} \\
 pm_{kh} &\geq 0
 \end{aligned} \tag{4.35}$$

#### 2.4.2. Contrainte d'unicité d'exécution

Cette contrainte est assurée par l'opérateur d'affectation binaire  $\alpha_{ijk}$ .

$$\alpha_{ijk} = \begin{cases} 1 & \text{si l'opération } O_{ij} \text{ est réalisé sur la machine } M_k, \\ 0 & \text{si non} \end{cases}$$

Pour que l'opérateur  $\alpha_{ijk}$  assure aussi, que toute opération  $O_{ij}$  est exécutée une fois au maximum par une seule machine  $M_k$ , il doit remplir la condition équation (4.36).

$$\sum_{\substack{1 \leq k \leq m \\ O_{ij} \in \Phi_k}} \alpha_{ijk} = 1 \quad \forall i, j \tag{4.36}$$

Les contraintes de l'inégalité (4.37) assurent l'attribution de l'opération  $O_{ij}$  à une seule machine.

$$c_{ij} - s_{ij} \geq \sum_{\substack{1 \leq k \leq m \\ O_{ij} \in \Phi_k}} (p_{ijk} \times \alpha_{ijk}) \tag{4.37}$$

#### 2.4.3. Contrainte d'enchaînement

Contrainte d'enchaînement inégalité (4.38) représente la relation de succession entre les différentes opérations d'un même job. C'est-à-dire, elle assure le respect de la gamme opératoire imposée par le bureau de méthode. Autrement dit l'opération  $O_{i(j+1)}$  doit succéder à l'opération  $O_{ij}$ .

$$\begin{aligned}
 O_{ij} &< O_{i(j+1)} \\
 s_{i(j+1)} - c_{ij} &\geq 0 \quad \forall i \in \{1,2, \dots, n\}, j \in \{1, 2, \dots, (n_i - 1)\} \tag{4.38}
 \end{aligned}$$

Ou, encore :

$$s_{i(j+1)} \geq s_{ij} + p_{ij} \quad \forall i \in \{1,2, \dots, n\}, j \in \{1, 2, \dots, (n_i - 1)\}$$

#### 2.4.4. Les contraintes de non chevauchement

Les contraintes de l'inégalité (4.39) et de l'inégalité (4.40) assurent le non chevauchement des opérations  $O_{ij}$  et  $O_{st}$ , qui s'exécutent sur la même machine. Les deux opérations ne peuvent s'exécuter en aucun cas en même temps sur la même machine. Ces opérations doivent respecter les

conditions suivantes : si  $O_{ij}$  et  $O_{st} \in \Phi_k$  de date de début  $s_{ij}$  et  $s_{st}$  respectivement et si  $O_{st}$  est sur la machine  $M_k$  alors  $s_{ij} \notin [s_{st}, s_{st} + p_{st}]$ .

L'enchaînement est organisé par la présence de l'opérateur  $\pi_{ijst}$ , défini suivant l'une des règles de priorités.

$$s_{ij} - s_{rt} + G[\pi_{ijrt} + (1 - \alpha_{ijk}) + (1 - \alpha_{rtk})] \geq p_{rt} \times \alpha_{rtk} \quad (4.39)$$

$$s_{rt} - s_{ij} + G[(1 - \pi_{ijrt}) + (1 - \alpha_{ijk}) + (1 - \alpha_{rtk})] \geq p_{ij} \times \alpha_{ijk} \quad (4.40)$$

$$\forall i, r = \{1, \dots, n\}, \forall j, t = \{1, \dots, n_i\}, \forall k = \{1, \dots, m\}$$

$$\forall O_{ij} \text{ et } O_{rt} \in \Phi_k$$

Avec

$$\pi_{ijrt} = \begin{cases} 1 & \text{si } O_{ij} < O_{st} \\ 0 & \text{si non} \end{cases}$$

$G$  est un nombre très grand, généralement égal à la somme des durées opératoire des opérations à ordonnancer, équation (4.41).

$$G = \sum_{1 \leq i \leq n} \sum_{1 \leq j \leq n_i} p_{ij} \quad (4.41)$$

La présence de  $(1 - \alpha_{ijk})$  et  $(1 - \alpha_{rtk})$  est pour s'assurer que les deux opérations  $O_{ij}$  et  $O_{rt}$  sont programmées sur la même machine. Dans le cas où,  $O_{rt}$  doit s'exécuter avant  $O_{ij}$  ( $O_{rt} < O_{ij}$ ),  $\pi_{ijrt} = 0$ , alors l'inégalité (4.40) est validée  $s_{ij} - s_{rt} \geq p_{rt}$ , par contre l'inégalité (39) reste toujours validée indépendamment des valeurs de  $s_{ij}$  et  $s_{rt}$  et inversement si ( $O_{ij} < O_{rt}$ ).

#### 2.4.5. Contrainte d'indisponibilité

Avec la présence des opérations de maintenance préventive systématique (périodes d'indisponibilité) les opérations de production ne peuvent pas être interrompues par d'autre opération, c'est le cas d'ordonnancement non-préemptif. C'est-à-dire si  $O_{ij}$  commence, elle doit achever sans interruption. Deux situations conflictuelles entre les opérations de production et les opérations de maintenance sont probables. Ces situations sont gérées par les contraintes d'indisponibilités suivantes, inégalité (4.42) et inégalité (4.43).

$$s_{ij} - sm_{kh} + M\delta_{ijkh} \geq pm_{kh} \quad \forall (Om_{kh}, O_{ij}) \in \Phi_k \quad (4.42)$$

$$sm_{kh} - s_{ij} + M(1 - \delta_{ijkh}) \geq p_{ij} \times \alpha_{ijk} \quad \forall (Om_{kh}, O_{ij}) \in \Phi_k \quad (4.43)$$

Dans la première situation, l'opération de maintenance commence avant l'opération de production  $Om_{kh} < O_{ij}$ . Face à cette situation :

- 1) Soit l'opération de production est maintenue, et l'opération de maintenance est avancée avec pénalité d'avance, Figure (4.4).

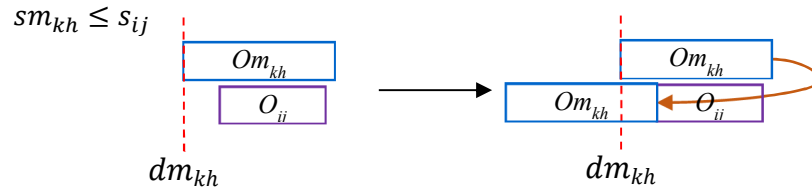


Figure (4.4) : cas de  $O_{i'j'} < Om_{kh}$  et  $Om_{kh} < O_{ij}$

Dans ce cas si  $O_{i'j'} < O_{ij}$  alors  $O_{i'j'} < Om_{kh}$  et  $Om_{kh} < O_{ij}$ , inégalité (4.44) et (4.45)

$$sm_{kh} \geq (s_{i'j'} + p_{i'j'}) \times \alpha_{i'j'k} \quad (4.44)$$

et

$$s_{ij} \times \alpha_{ijk} \geq sm_{kh} + pm_{kh} \quad \text{avec} \quad dm_{kh} \geq sm_{kh} \quad (4.45)$$

- 2) Soit l'opération de maintenance est maintenue, et l'opération de production est retardée avec pénalité de retard, Figure (4.5)

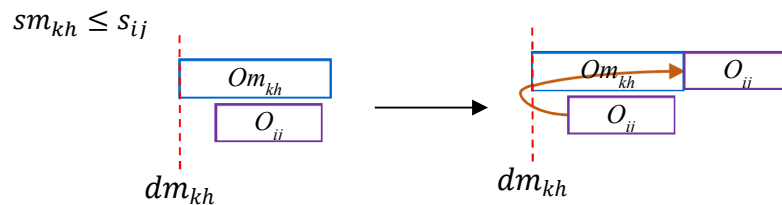


Figure (4.5) : cas de  $Om_{kh} < O_{ij}$

Dans ce cas  $Om_{kh} < O_{ij}$ , Inégalité (4.46).

$$s_{ij} \times \alpha_{ijk} \geq sm_{kh} + pm_{kh} \quad \text{avec} \quad sm_{kh} = dm_{kh} \quad (4.46)$$

Dans la deuxième situation, l'opération de la production commence avant l'opération de la maintenance  $O_{ij} < Om_{kh}$ . Devant cette situation :

- 3) Soit l'opération de production est maintenue, et l'opération de maintenance est retardée avec pénalité de retard, Figure (4.6).

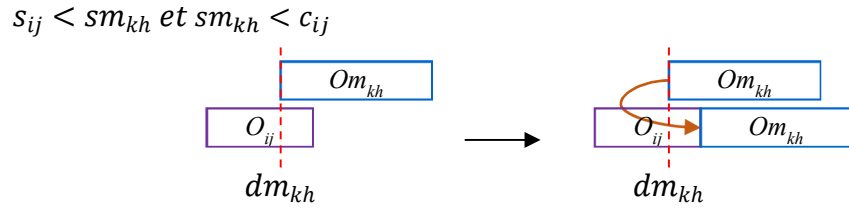


Figure (4.6) : cas de  $O_{ij} < Om_{kh}$  avec retard de  $Om_{kh}$

Cette situation est gérée par la contrainte, inégalité (4.47).

$$sm_{kh} \geq (s_{ij} + O_{ij}) \times \alpha_{ijk} \text{ avec } sm_{kh} \geq dm_{kh} \quad (4.47)$$

- 4) Soit l'opération de maintenance est maintenue, et l'opération de production est avancée avec pénalité d'avance, Figure (4.7).

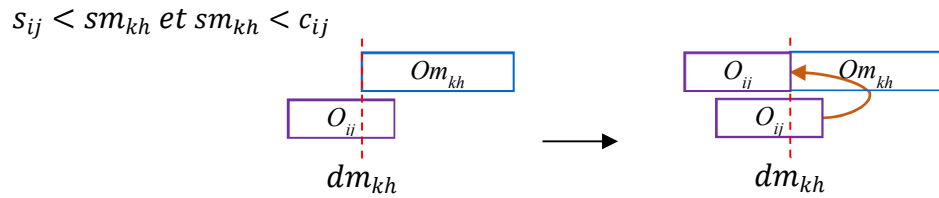


Figure (4.7) : cas de  $O_{ij} < Om_{kh}$  avec avance de  $O_{ij}$

Cette situation est gérée par la contrainte, inégalité (4.48).

$$sm_{kh} \geq (s_{ij} + O_{ij}) \times \alpha_{ijk} \text{ avec } sm_{kh} = dm_{kh} \quad (4.48)$$

### 3. Application de NSGA-II au MCJSSP

Nous présentons dans cette rubrique une description de l'algorithme proposé pour la résolution du MCJSSP, les opérateurs utilisés par NSGA-II sont aussi détaillés.

#### 3.1. Description de l'algorithme NSGA-II

Le NSGA-II est l'un des algorithmes évolutionnaires les plus populaires pour l'optimisation multiobjectif. La section 5.1.2 et 5.2.5 chapitre 3 en décrit le fonctionnement.

L'évolution de l'algorithme est basée sur quatre opérateurs : la sélection, le croisement, la mutation et l'évaluation. L'opérateur de sélection choisit les parents pour le croisement. Le croisement échange les informations entre individus pour en créer de nouveau, ceci permet à l'algorithme de converger. La mutation modifie des individus nouvellement créés par le croisement des parents, permet ainsi à l'algorithme d'éviter les optimums locaux. L'évaluation des individus donne une note à chaque individu et estime la diversité de la population.

Une nouvelle population est sélectionnée suivant la complémentarité entre les critères :

- 1) Les parents et les enfants sont réunis dans la même population et sont classés en fronts de Pareto, suivant le principe de dominance. Nous obtenons ainsi, un ensemble de rang  $R = \{R_1, R_2, \dots, R_m\}$ , où :

$$\bigcup_{1 \leq i \leq m} R_i = P \quad \text{et} \quad \bigcap_{1 \leq i \leq m} R_i = \emptyset$$

Ces fronts sont classés suivant le degré de dominance. Le front de rang  $R_1$  est le front dominant tant dis que, celui du rang  $R_m$  est le front dominé.

- 2) À l'intérieur de chaque front de Pareto, la densité d'une solution particulière (de chaque individu) est estimée, par le calcul de la distance entre la solution particulière et les solutions qui lui sont les plus proches. En additionnant les distances ainsi calculées pour chaque critère on obtient une distance dite de crowding.

Le principe du crowding distance est d'étirer le front de Pareto optimal. Pour cela, les solutions extrêmes du front de Pareto sont privilégiées dans la notation ainsi que les individus les plus éloignés de leur voisin. Le crowding distance est calculé pour chaque individu.

La nouvelle population est complétée en partant des individus du meilleur front de Pareto, jusqu'à atteindre la limite de population. Si, en ajoutant les individus d'un front, la limite de population est atteinte, le choix des individus est déterminé par leur note obtenue par le crowding distance. Le principe de fonctionnement de l'algorithme NSGA-II se résume par son Pseudo-code, Algorithme (4.1).

Initialiser les populations  $P_0$  et  $Q_0$  de taille  $N$

**Tant que** critère d'arrêt non rencontré faire

Création de  $R_t = P_t \cup Q_t$

Calcul des différents fronts  $F_i$  de la population  $R_t$  par un algorithme de « ranking »

Mettre  $P_{t+1} = \emptyset$  et  $i = 0$

**Tant que**  $|P_{t+1}| + |F_i| < N$  faire

-  $P_{t+1} = P_t \cup F_i$

-  $i = i + 1$

**Fin Tant Que**

Inclure dans  $P_{t+1}$  les  $(N - |P_{t+1}|)$  individus de  $F_i$  les mieux répartis au sens de la « distance de crowding »

Sélection dans  $P_{t+1}$  et création de  $Q_{t+1}$  par application des opérateurs de croisement et mutation

**Fin Tant Que**

Algorithme (4.1) : Pseudo-code de NSGA-II

### 3.2. Les opérateurs proposés pour NSGA-II

Les algorithmes NSGA-II, décrits en détail dans le quatrième chapitre de cette thèse, sont connus pour leur capacité à fournir de bonnes solutions aux problèmes d'optimisation combinatoire NP-difficiles, en un temps relativement court. En particulier, ils ont prouvé leur efficacité dans la résolution de problèmes d'ordonnancement multicritères. Pour la résolution du problème d'ordonnancement job shop multicritères intégré des activités de la production et de la maintenance, nous proposerons l'application de l'algorithme NSGA-II,

Les différentes étapes de l'algorithme NSGA-II, que nous proposons sont illustrées dans le pseudocode Figure (5.8) et seront détaillées dans les sous-sections suivantes. L'algorithme commence par l'initialisation de la génération initiale de manière aléatoire. Après, un processus d'ajustement est effectué sur chaque individu afin de satisfaire les différentes contraintes entre les différentes tâches ou jobs. Chaque individu sera évalué et classé en utilisant les trois critères. Le processus d'évolution de la population est assuré par différents opérateurs génétiques, notamment, la sélection, le croisement, la mutation et le remplacement. L'algorithme itère le processus jusqu'à atteindre un nombre maximum d'itérations ou jusqu'à ce que la valeur globale de fitness ne s'améliore plus.

#### 3.2.1. Codage des solutions

Le codage est une étape importante dans l'élaboration d'un algorithme génétique. Dans ce travail, nous adoptons le codage basé sur les opérations. Par ce codage ordonnancement est représenté comme une suite d'opérations. Chaque opération est codée par un gène, de deux manières possibles : soit l'opération est désignée par un symbole qui la distingue des autres, cas du codage utilisé dans le problème du voyageur de commerce, ce type est moins fiable pour le JSSP, [Ono et al. 1996]. Soit, toutes les opérations d'un même job portent le même symbole généralement le numéro du job. Avec ce codage, pour un problème de ( $n$ ) jobs et ( $m$ ) machine, le chromosome comprend  $\sum_{1 \leq i \leq n} n_i$  gènes. Chaque job ( $J_i$ ) apparaît dans le chromosome exactement ( $n_i$ ) fois (nombre d'opérations). L'ordre d'occurrence ( $k$ ) d'un job désigne sa ( $k^{ième}$ ) opération. Ce codage représente l'avantage de d'exiger qu'un simple constructeur d'ordonnancement [Bierwirth, 1995] et de couvrir l'espace de tous les ordonnancements actifs possibles, même si elle génère une certaine redondance.

Dans notre cas d'étude, le chromosome contient deux types de gènes : Les premiers gènes d'un chromosome représentent l'ordonnancement de la production. Ils sont de nombre d'opération à ordonnancer. Les ( $m$ ) gènes suivant représentent les opérations de maintenance respectivement

sur les machines  $M_k$   $k = \{1, 2, \dots, m\}$ . La longueur du chromosome est donc égale à  $(n + m)$  avec  $n$ : nombre d'opérations et  $m$  : nombre de machines.

Pour illustrer ce codage, on prend à titre d'exemple un problème de job shop à  $n = 3$  jobs constitué respectivement de 3, 4 et 3 opérations et  $m = 4$  machines dont chaque machine doit subir une opération de maintenance. Une contrainte à respecter est le non chevauchement des opérations de maintenance. Les jobs de la production sont numérotés de 1 à  $n$  et ceux de la maintenance de  $(n + 1)$  à  $(n + m)$ . Nous appliquons ci-dessous les deux étapes de codage. Les jobs et les tâches de maintenance sont codés comme sur la Figure (4.8).

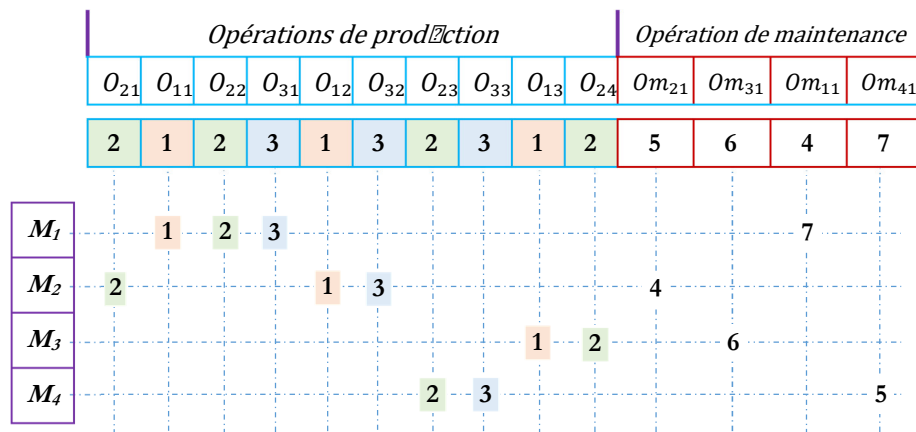


Figure (4.8) : Construction d'ordonnancement à partir d'un chromosome codé à base d'opérations

La deuxième étape de l'algorithme génétique étant la génération de la population initiale. C'est l'objet du paragraphe suivant.

### 3.2.2. Génération aléatoire de la population initiale

Le choix de la population initiale conditionne fortement la rapidité de l'algorithme. Dans la littérature, il existe plusieurs méthodes de génération de la population initiale : génération aléatoire, génération par duplication et évolution ou en s'appuyant sur une heuristique. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus, en veillant à ce que les individus produits respectent les contraintes. Si par contre, des informations a priori sur le problème sont disponibles, il paraît naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence. Dans la suite de notre travail nous utilisons la génération aléatoire de la population initiale, Algorithme (4.2).

```

Début
Entrées
    Taille de la population ;
    Nombre d'opération à ordonnancer ;
    Nombre de machines ;
    Nombre d'opération de maintenance ;
Pour i de 1 jusqu'à la taille de la population Faire
    Générer aléatoirement un chromosome ;
    Si le chromosome est cohérent Alors
        Ajouter le chromosome à la population initiale
    Sinon
         $i \leftarrow i - 1$  ;
    fin si
Fin Pour
Sorties
    Population initiale
Fin

```

Algorithme (4.2) : Population initiale

### 3.2.3. Opérateurs génétiques

Les opérateurs génétiques représentent le cœur d'un algorithme génétique. L'utilisation de tels opérateurs est nécessaire pour l'évolution et la génération de nouvelles solutions à partir des populations précédentes.

#### 3.2.3.1. Opérateurs de sélection et de croisement

L'opérateur de croisement est utilisé pour générer une ou deux nouvelles solutions à partir de deux parents. Toutefois, avant d'effectuer le croisement, nous utilisons un processus de sélection afin de choisir les parents aptes à la reproduction. Ce choix est fait suivant l'algorithme Algorithme (4.3).

```

Début
Entrées
    Taille de la population ;
    Taille de tournoi ;
    Distance de crowding ;
    Rang des fronts de Pareto ;
Pour i de 1 jusqu'à taille de la population Faire
    Sélectionnez n (= taille du tournoi) individus au hasard.
    Pour j de 1 la taille de tournoi Faire
        Candidat (j) ← Sélection aléatoire d'un individu
        S'assurer que les candidats identiques ne sont pas choisis
    Fin pour
    Collecte des informations sur les candidats sélectionnés.
    Trouver le candidat avec le plus petit rang
    Si plus d'un candidat ont le plus petit rang, alors, choisir le candidat au
    sein de ce groupe ayant la distance de crowding maximale.

```



Si quelques individus ont le plus petit rang et une distance de crowding maximale, sélectionner un seul individu au hasard  
 Ajouter l'individu sélectionné à l'ensemble des candidats à accouplement  
**Fin pour**  
 On obtient l'ensemble des individus sélectionnés pour la reproduction.  
**Fin**

Algorithme (4.3) : Algorithme de sélection

### 3.2.3.2. Opérateurs de croisement

L'objectif d'un opérateur de croisement est de combiner les caractéristiques de deux bonnes solutions afin de générer des solutions encore meilleures. Il existe plusieurs types de croisements. Dans cette thèse, après plusieurs essais sur différents types de croisement, nous proposons l'opérateur de croisement d'ordre linéaire (LOX), pour la séquence des opérations de la production. Le principe d'implémentation de LOX est exprimé par l'algorithme Algorithme (4.4).

**Algorithme** le croisement LOX

**Entrées :**

Liste des chromosomes candidats à la reproduction, composée de  $T_p$  chromosomes, de longueur  $m + N$   $m$  : nombre d'opération de maintenance et  $N = \sum_{1 \leq i \leq n} n_i$  nombre d'opération de production.

Choix aléatoire deux chromosomes parents dans la liste des candidats à la reproduction :

$k_1 = \text{random}(1, T_p)$  ;

$k_2 = \text{random}(1, T_p)$  avec  $k_1 \neq k_2$  ;

Retirer les chromosomes  $k_1$  et  $k_2$  de la liste des candidats,  $T_p \leftarrow T_p - 2$

Mise à jour de la liste des candidats.

$P(1, N) \leftarrow P(k_1, N)$  et  $P(2, N) \leftarrow P(k_2, N)$

Choisir aléatoirement la longueur de la sous-séquence  $S_s$  :  $l = \text{random}\left(1, \frac{n}{2}\right)$  ;

Choisir aléatoirement la position

$d_1 = \text{random}\left(1, \frac{n}{2}\right)$  ;

$d_2 = d_1 + l$

Copier  $S_s$  entre  $d_1$  et  $d_2$  de  $P(1, N)$  et la collée dans la même position dans le chromosome enfant  $E(2, N)$  ;

Copier  $S_s$  entre  $d_1$  et  $d_2$  de  $P(2, N)$  et la collée dans la même position dans le chromosome enfant  $E(1, N)$  ;

**Pour**  $i$  de 1 jusqu'à 2 **Faire**

**Si**  $i=1$  **Alors**

$j=2$  ;

**Sinon**  $j=1$

**Fin Si**

**Pour**  $k$  de 1 jusqu'à  $N$  **Faire**

**Si**  $P(i, k) \notin S_s$  de  $E(j, N)$  **Alors**

**Si**  $u < d_1$  **Alors**

$E(u, N) \leftarrow P(i, k)$  ;

```

        u = u + 1 ;
    Sinon
        u = d2 + 1 ;
    Fin Si
Fin Pour
Fin Pour
Sortie
Liste des chromosomes enfants.
    
```

## Algorithme (4.4) : principe de croisement LOX

Pour la séquence des opérations de maintenance, nous proposons le croisement dont l'implémentation est présentée dans l'Algorithme (4.5). Il est à noter que, les chromosomes dans l'algorithme de l'Algorithme (3.4) sont codés par numéros des opérations. Ce qui fait, le codage des enfants obtenus doit être transformé en codage par les jobs, pour éviter les corrections.

**Algorithme** le croisement par croisement

**Entrées :**

Liste des chromosomes candidats à la reproduction, composée de  $T_p$  chromosomes, de longueur  $m + N$   $m$  : nombre d'opération de maintenance et  $N = \sum_{1 \leq i \leq n} n_i$  nombre d'opération de production. Choix aléatoire deux chromosomes parents dans la liste des candidats à la reproduction :

$k_1 = \text{random}(1, T_p)$  ;

$k_2 = \text{random}(1, T_p)$  avec  $k_1 \neq k_2$  ;

Retirer les chromosomes  $k_1$  et  $k_2$  de la liste des candidats,  $T_p \leftarrow T_p - 2$

Mise à jour de la liste des candidats.

$P(1, N) \leftarrow P(k_1, N)$  et  $P(2, N) \leftarrow P(k_2, N)$

$d_1 = N + \text{partie entière de } \left(\frac{m}{2}\right)$

$d_2 = N + 1 + \text{partie entière de } \left(\frac{m + 1}{2}\right)$

$E(2, d_1) \leftarrow P(1, N + m)$

$E(2, d_2) \leftarrow P(k_2, N + 1)$

$E(1, N + 1) \leftarrow P(2, d_2)$

$E(2, N + m) \leftarrow P(k_2, d_1)$

**Pour** i de 1 jusqu'à 2 **Faire**

**Si** i=1 **Alors**

        j=2 ;

**Sinon** j=1 ;

**Fin Si**

**Pour** k de 1 jusqu'à N **Faire**

**Si**  $P(i, k) \notin E(j, [N + 1, N + m])$  **Alors**

**Tant que**  $E(j, u)$  **et plain faire**

$u = u + 1$  ;

**Fin tant que**

$E(j, u) \leftarrow P(i, k)$  ;

**Fin Pour**

**Fin Pour**  
**Sortie**  
 Liste des chromosomes enfants.

Algorithme (4.5) : principe de croisement pour maintenance par croisement

Pour illustrer le mécanisme de croisement combiné production et maintenance, nous présentons l'exemple suivant : Figure (4.9).

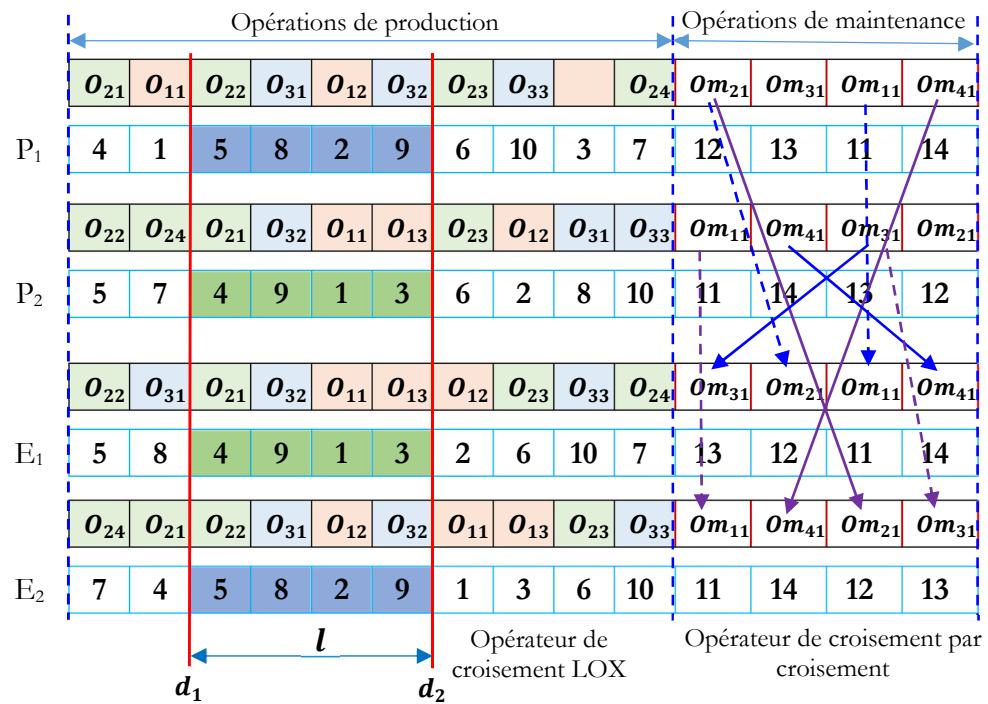


Figure (4.9) : croisement combiné production maintenance

### 3.2.3.3. Opérateurs de mutation

L'objectif visé par l'application de l'opérateur de mutation est de porter une certaine diversité dans la population, pour empêcher que celle-ci converge trop vite vers un optimal local, et d'éviter aussi, de se coincer dans ce dernier. La mutation est réalisée en modifiant un gène d'un individu pris au hasard. Dans les Algorithmes Génétiques, la mutation est considérée comme un opérateur secondaire par rapport au croisement. Dans cette thèse la mutation par décalage figure (4.10).

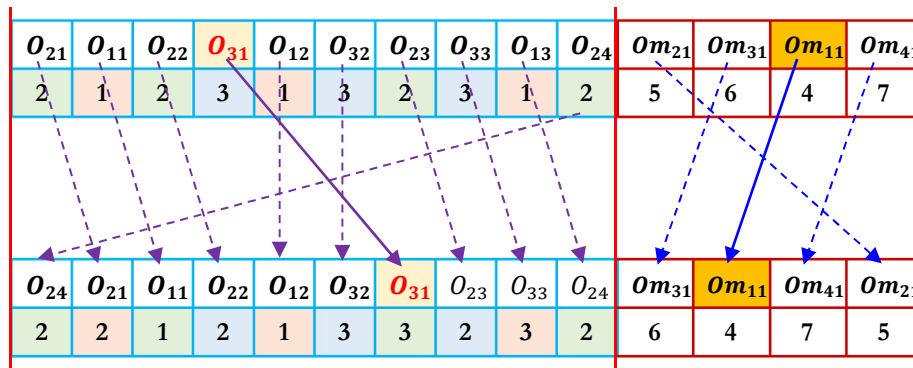


Figure (4.10) : Mutation basée sur permutation des jobs

### 3.3. Expérimentations et Résultats

Cette section présente les résultats des évaluations conduites par l'approche proposée pour la résolution du problème optimisation multicritères de l'ordonnancement des Activités de la Production et de la Maintenance Intégrées dans un Atelier Job Shop.

L'évaluation des performances de notre approche est concrétisée, sur un ensemble d'instances JSSP de référence décrite dans [CSP2SAT, 2007]. En raison de l'absence de travaux relatifs qui considèrent, les trois métriques de qualité de service Makespan, coût de production et coût de maintenance, en même temps, nous avons comparé nos résultats avec d'autre travaux pour le critère de Makespan, en suite, les résultats obtenus des trois critères avec et sans prise en considération de la maintenance.

[CSP2SAT, 2007] contient plusieurs types d'instance rectangulaire de JSSP, de différentes tailles. Ces instances sont caractérisées, par le passage de tous les jobs sur toutes les machines  $\{M_k\}_{k=1}^{k=m}$ . Parce que les notations et les formules se simplifient pour ce type d'instance, il est assez courant dans la littérature que les algorithmes ne traitent que les instances rectangulaires. Pour ces instances on a  $(n \times m)$  opérations à ordonnancer et toutes les machines doivent traiter tous les opérations une et une seule fois. Les données de ces instances sont représentées par une matrice rectangulaire (d'où la qualification attribuée). Dans notre algorithme, pour traiter les cas où, les jobs ne passent pas tous sur tous les machines, les opérations non existantes sont remplacées par un zéro, de même pour les machines correspondantes. Les instances de référence sont regroupées en classes. Nous avons utilisé, les classes : de [Laurent, 1984] (de La01 à La32), de [Fisher et Thompson 1963] (ft6, ft10 et FT20) et enfin de [Applegate et Cook 1991] (de orb1 à obr9).

Nous avons exploité l'environnement de programmation MATLAB, pour programmer l'algorithme NSGA-II. Notre programme est organisé en un programme principal et sept sous-programmes : population initial, tournoi de sélection, Croisement et Mutation, Non Domination,

satisfaction des contraintes, substitution des chromosomes et évaluation critères. Tous nos tests ont été exécutés sur un PC avec un processeur Intel Pentium IV à 2,00 GHz CPU et 01 Go de RAM, avec un système d'exploitation Windows dotés de la version 32 bits.

Les paramètres utilisés pendant le processus expérimental ont été choisis, suite à une série d'expériences préliminaires : la taille de la population  $T_p$  est calculée par la formule empirique  $T_p = m \times n + 100$  où  $m$  : nombre de machines et  $n$  : nombre de jobs. La probabilité de croisement est de 68% et la probabilité de mutation est de 1,5%. Le nombre de générations qui représente le critère d'arrêt est défini en fonction des instances entre 200 et 400, La plupart des 44 instances adoptés ont besoin de 250 itérations chacune, à l'exception de quelques-unes ont besoin de 400 itérations. À l'issu de 10 expériences indépendantes sur chaque instance. Les résultats obtenus sont regroupés dans le Tableau (4.1).

### 3.3.1. Comparaison du Makespan

Pour s'assurer des performances de notre algorithme, nous comparons les valeurs du Makespan trouvés par notre algorithme (en considérant le critère de Makespan seul), d'une part avec les meilleures solutions connues (BKS), d'autre part avec les valeurs de référence (BKR) trouvées par les recherches les plus récentes : Mohsen (2014), Nilsen et al. (2016), Mohamed (2016).

Comme, le montre le schéma de la Figure (4.11), les Makespan de BKS sont relativement inférieurs aux makespan1 de BKR et aux makespan2 trouvés par notre algorithme. Cette légère différence montre l'importance de nos résultats.

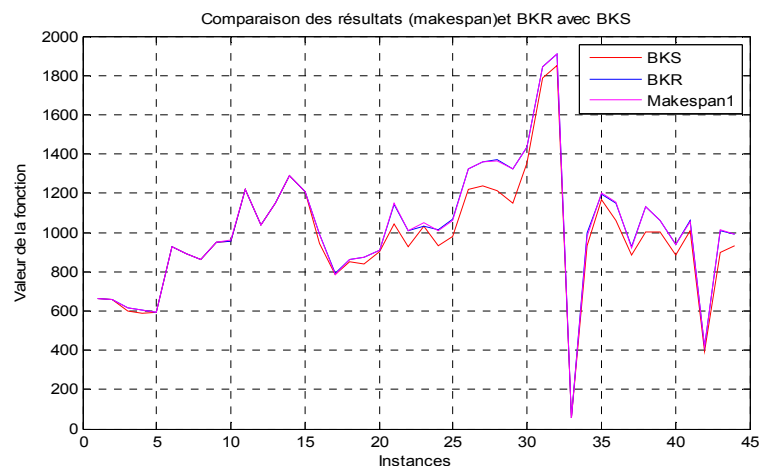


Figure (4.11) : Comparaison des résultats (Makespan) avec BKS

Par contre, nos valeurs de Makespan sont très proches de BKR, ce qui justifie la performance de notre algorithme. Pour mesurer les performances de nos algorithmes, nous avons calculé les ratios de la déviation relative, que nous appelons RPD entre les meilleures solutions obtenues par

l'algorithme et les meilleures solutions connues (BKS). Ainsi que la déviation de nos makespan par rapport aux BKR. Après tous calculs faits par l'équation (47), les déviations sont présentées par la Figure (4.12).

N <sup>o</sup>	Nom Instance	Size		REF-1			Nos résultats d'optimisation									
		n	m	BKS	BKR	RPD <sub>1</sub>	Un critère		Deux critères			Trois critères				
							Makespan 1	RPD <sub>2</sub>	Makespan 2	coût production	RPD <sub>3</sub>	Makespan 3	coût production	coût maintenance	RPD <sub>4</sub>	RPD <sub>5</sub>
1	la01	10	5	666	666	0.00	666	0.00	851	3308	27.78	855	3480	706	28.38	5.20
2	la02	10	5	655	655	0.00	655	0.00	894	3197	36.49	899	3380	756	37.25	5.70
3	la03	10	5	597	617	3.35	619	3.69	864	2754	44.72	867	2889	662	45.23	4.90
4	la04	10	5	590	607	2.88	605	2.54	741	2629	25.59	747	2763	590	26.61	5.10
5	la05	10	5	593	593	0.00	596	0.51	752	2702	26.81	756	2829	565	27.49	4.70
6	la06	15	5	926	926	0.00	928	0.22	1296	4260	39.96	1304	4524	945	40.82	6.20
7	la07	15	5	890	890	0.00	890	0.00	1136	4228	27.64	1142	4499	1005	28.31	6.40
8	la08	15	5	863	863	0.00	863	0.00	1147	4263	32.91	1156	4540	939	33.95	6.50
9	la09	15	5	951	951	0.00	951	0.00	1209	4546	27.13	1216	4810	957	27.87	5.80
10	la10	15	5	958	958	0.00	962	0.42	1210	4322	26.30	1218	4569	946	27.14	5.70
11	la11	20	5	1222	1222	0.00	1222	0.00	1556	5846	27.33	1563	6243	1386	27.91	6.80
12	la12	20	5	1039	1039	0.00	1040	0.10	1409	4824	35.61	1420	5143	1165	36.67	6.60
13	la13	20	5	1150	1150	0.00	1150	0.00	1413	5636	22.87	1420	6042	1324	23.48	7.20
14	la14	20	5	1292	1292	0.00	1292	0.00	1680	5852	30.03	1689	6285	1232	30.73	7.40
15	la15	20	5	1207	1207	0.00	1210	0.25	1503	5457	24.52	1511	5882	1275	25.19	7.80
16	la16	10	10	945	994	5.19	991	4.87	1205	4244	27.51	1214	4494	962	28.47	5.90
17	la17	10	10	784	793	1.15	787	0.38	975	3630	24.36	985	3898	875	25.64	7.40
18	la18	10	10	848	860	1.42	861	1.53	1192	4071	40.57	1197	4347	965	41.16	6.80
19	la19	10	10	842	873	3.68	875	3.92	1175	4192	39.55	1184	4481	899	40.62	6.90
20	la20	10	10	902	912	1.11	911	1.00	1168	4050	29.49	1176	4335	882	30.38	7.05
21	la21	15	10	1046	1146	9.56	1148	9.75	1366	5064	30.59	1372	5475	1131	31.17	8.10
22	la22	15	10	927	1007	8.63	1011	9.06	1236	4626	33.33	1241	4991	1029	33.87	7.90
23	la23	15	10	1032	1033	0.10	1047	1.45	1377	5211	33.43	1389	5643	1113	34.59	8.30
24	la24	15	10	935	1012	8.24	1009	7.91	1246	4487	33.26	1254	4877	1097	34.12	8.70
25	la25	15	10	977	1067	9.21	1059	8.39	1450	4454	48.41	1461	4859	1059	49.54	9.10
26	la26	20	10	1218	1323	8.62	1326	8.87	1594	5577	30.87	1602	6001	1354	31.53	7.60
27	la27	20	10	1235	1359	10.04	1359	10.04	1638	6068	32.63	1644	6596	1343	33.12	8.70
28	la28	20	10	1216	1369	12.58	1368	12.50	1650	5799	35.69	1661	6211	1368	36.60	7.10
29	la29	20	10	1152	1322	14.76	1325	15.02	1788	5245	55.21	1800	5633	1094	56.25	7.40
30	la30	20	10	1355	1437	6.05	1434	5.83	1838	6337	35.65	1846	6876	1371	36.24	8.50
31	la31	30	10	1784	1844	3.36	1845	3.42	2568	8285	43.95	2578	9048	1760	44.51	9.20
32	la32	30	10	1850	1907	3.08	1909	3.19	2657	9196	43.62	2665	10190	2186	44.05	10.80
<b>33</b>	<b>ft06</b>	<b>6</b>	<b>6</b>	<b>55</b>	<b>55</b>	<b>0.00</b>	<b>55</b>	<b>0.00</b>	<b>58</b>	<b>1174</b>	<b>5.45</b>	<b>68</b>	<b>1284</b>	<b>217</b>	<b>10.91</b>	<b>5.30</b>
34	ft10	10	10	930	997	7.20	978	5.16	1371	4542	47.42	1381	4824	982	48.49	6.20
35	ft20	20	5	1165	1196	2.66	1201	3.09	1662	5592	42.66	1671	5922	1301	43.43	5.90
36	orb01	10	10	1059	1149	8.50	1156	9.16	1447	5479	36.64	1461	5868	1160	37.96	7.10
37	orb02	10	10	888	929	4.62	919	3.49	1192	4412	34.23	1205	4711	961	35.70	6.80

38	orb03	10	10	1005	1129	12.34	1131	12.54	1557	5568	54.93	1575	5913	1329	56.72	6.20
39	orb04	10	10	1005	1062	5.67	1064	5.87	1446	4771	43.88	1455	5133	1119	44.78	7.60
40	orb05	10	10	887	936	5.52	938	5.75	1249	4352	40.81	1260	4660	973	42.05	7.10
41	orb06	10	10	1010	1060	4.95	1058	4.75	1384	4994	37.03	1392	5364	1135	37.82	7.40
42	orb07	10	10	397	416	4.79	421	6.05	558	2056	40.55	558	2222	497	40.55	8.10
43	orb08	10	10	899	1010	12.35	1016	13.01	1408	4809,7	56.62	1410	5170	1083	56.84	7.50
44	orb09	10	10	934	994	6.42	989	5.89	1191	4602,2	27.52	1194	4920	1109	27.84	6.90
<b>Moyenne</b>						<b>4.27</b>		<b>4.31</b>			<b>35.04</b>				<b>35.95</b>	<b>7.04</b>

Tableau (4.1) : résultats des expériences

$$RPD_i = \frac{Makespan_i - BKS}{BKS} \times 100 \quad i = \{1,2\} \quad (47)$$

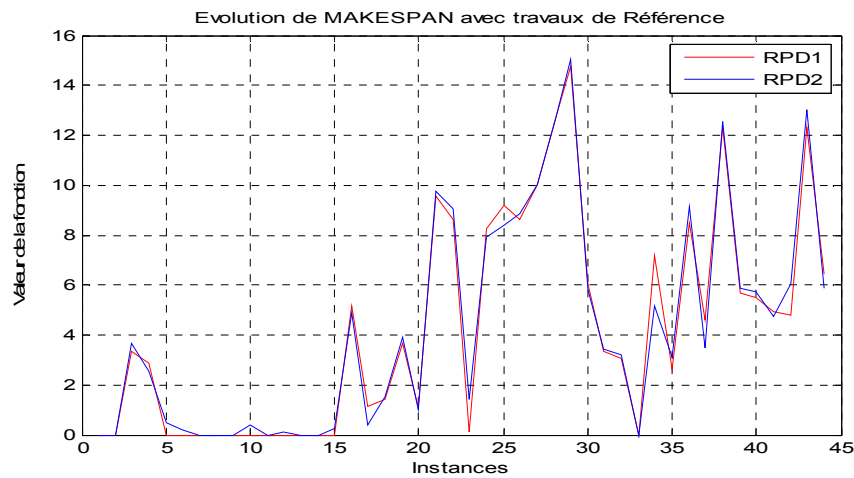


Figure (4.12) : Comparaison des gains résultats RPD1 et RPD2

La déviation moyenne, est très faible et ne représente que 4.27% pour  $RPD_1$  et 4.31% pour  $RPD_2$ . Ces résultats montrent la cohérence des résultats trouvés par l’algorithme proposé avec les résultats récemment publiés.

Dans la suite des tests, nous nous concentrons sur l’interprétation des résultats d’optimisation du Makespan et les coûts de production.

### 3.3.2. Analyse du Makespan et des coûts de production

Dans cette deuxième étape, les Makespan obtenu par optimisation intégrée des deux critères : Makespan et coûts de production, sont comparés, une fois au BKS et une deuxième fois au makespan1. Les déviations correspondantes sont importantes (RPD3 et RPD4), de valeur moyenne 35.04% pour la première et de 29.45% pour la seconde. Ceci est visible dans la Figure (4.13). Néanmoins, Ces valeurs sont justifiées par le passage du problème d’optimisation monocritère au problème d’optimisation multicritères, où le Makespan prend les valeurs de compromis. Les résultats sur la Figure (4.13) représentent les meilleurs résultats après 10 expériences sur chaque instance.

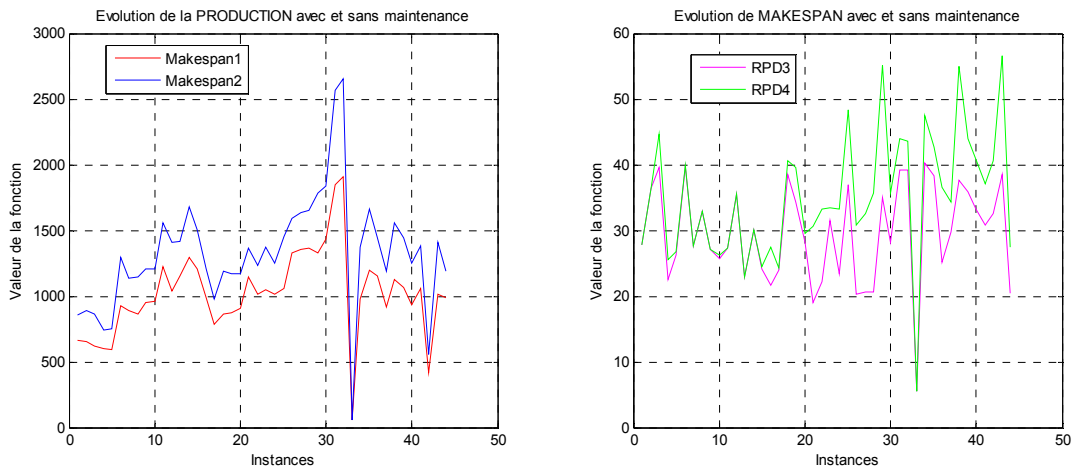


Figure (4.13) : Comparaison des déviations RPD3 et RPD4

### 3.3.3. Analyse du Makespan, coûts de production et coûts de maintenance

Dans cette section, les différents résultats de l'optimisation multicritères de l'ordonnement job shop intégrée des activités de production et de maintenance sont obtenus par l'utilisation de l'algorithme NSGA-II proposé. Ces résultats sont présentés dans le tableau (5.1). Le but de l'analyse des résultats obtenus est de prouver la performance de l'algorithme proposé, et de s'assurer de sa capacité d'offrir des meilleures solutions de compromis optimale, minimisant simultanément les trois critères d'optimisation.

La Figure (4.14) présente une comparaison entre les valeurs des makespan obtenues avec sans prise en considération des contraintes d'indisponibilités pour maintenance préventive. Ces différents résultats montrent que les valeurs de makespan de la plupart des instances n'ont pas augmentés énormément. En moyenne cette augmentation est de 0.7027% avec un écart type de 1.5792, ce qui prouve que l'algorithme NSGA-II est capable d'absorber l'effet des périodes d'indisponibilité, par l'exploitation des périodes de non productivité des machines.

La Figure (4.15) montre l'accroissement des coûts de production lorsque la maintenance est prise en considération dans l'ordonnement. L'augmentation est en moyenne de 7.0352% avec écart type de 14.1256. Cette croissance est expliquée par l'exploitation de la flexibilité des opérations de maintenance, et du caractère juste à temps des opérations de production, qui pénalise toutes opérations en avance ou en retard, par rapport à la date d'échéance.



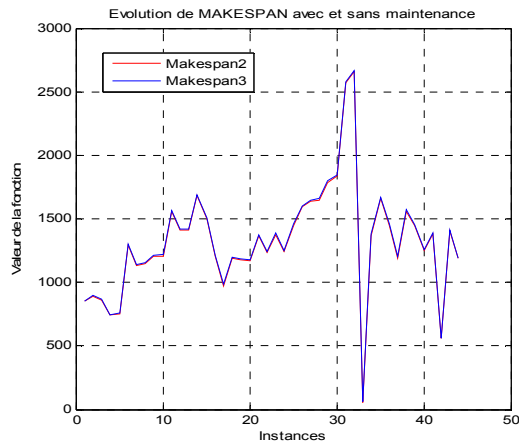


Figure (4.14) : évolution du Makespan sans intégration des opérations de maintenance.

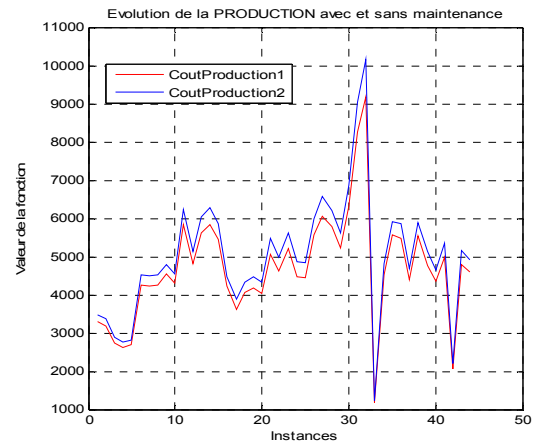


Figure (4.15) : évolution des coûts de production avec l'intégration des opérations de maintenance.

Les pourcentages de déviation relative du Makespan et des coûts de production sont présentés dans la Figure (4.16).

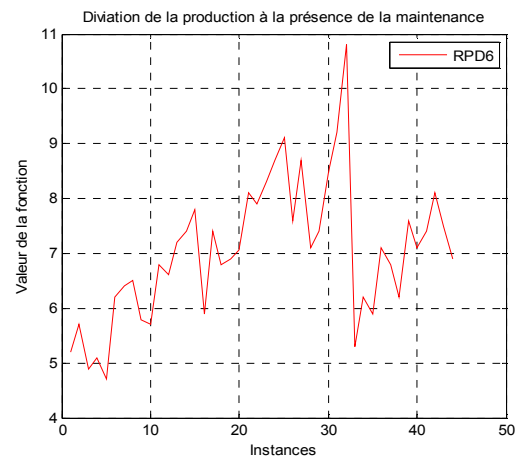
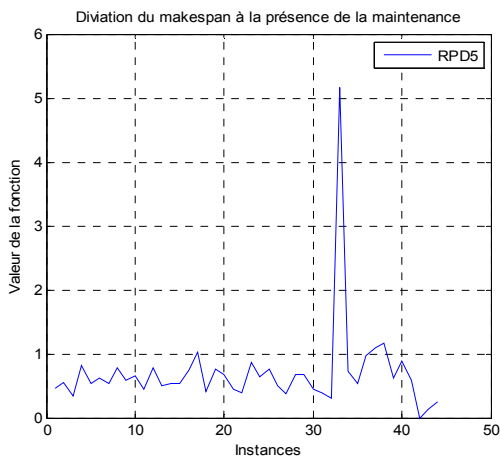


Figure (4.16) : déviation de makespan et production avec intégration des opérations de maintenance.

Dans ce qui suit, le benchmark de Muth et Thomas Mt06 sous contrainte d'indisponibilité est proposé comme exemple illustratif, pour analyser d'avantage l'évolution simultanée des critères retenus, à savoir le Makespan, coût de production et coût de maintenance, définis par les équations 1, 14 et 32.

La Figure (4.17), reproduit l'espace de solutions (Front de Pareto) représentant l'ensemble des individus (solutions) non dominés obtenu après plusieurs essais sur 400 générations d'une population de 136 individus.

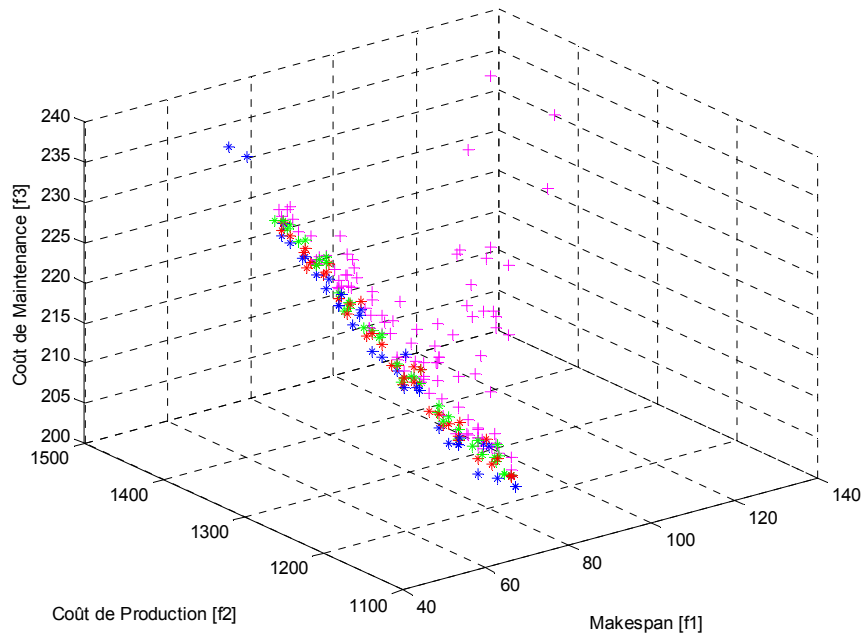


Figure (4.17) : les solutions Pareto optimales des critères obtenues par NSGA-II.

Les courbes obtenues par la Figure (4.18) reflètent l'adéquation des résultats obtenus par l'algorithme proposé. Pour chaque critère trois courbes sont présentées : la valeur de compromis du critère insérée entre la valeur maximale et la valeur minimal que peut prendre le critère.

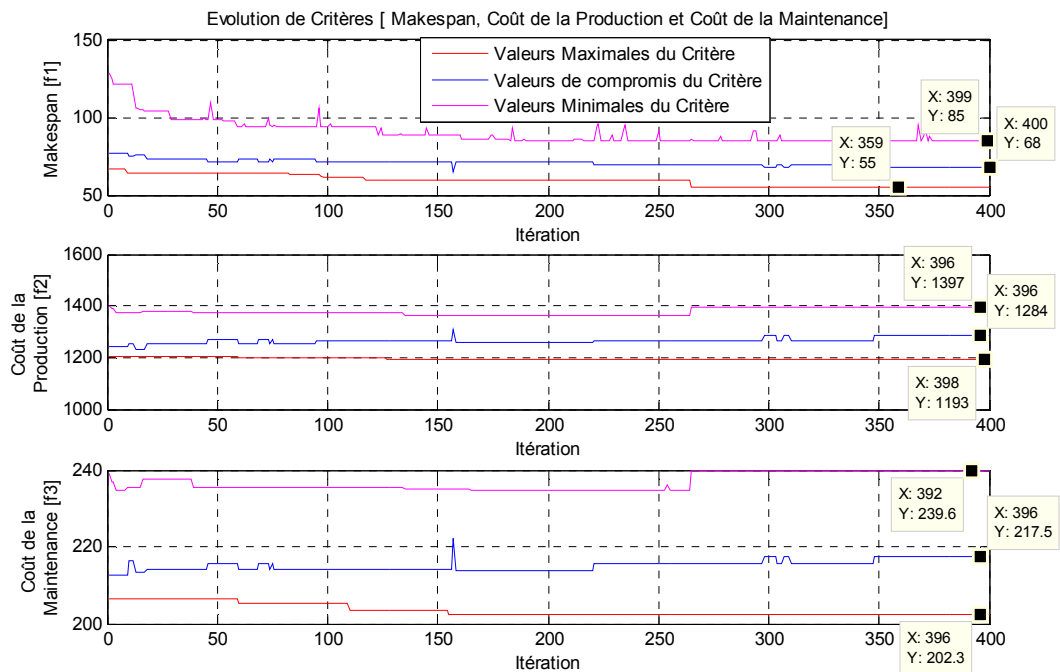


Figure (4.18): evolution des résultats obtenus par notre algorithme NSGA-II.

La Figure (4.19) présente le diagramme de Gantt correspondant à la même instance pour une solution de compromis optimale entre les trois critères où les rectangles en noire représentent l'ensemble des opérations du job de maintenance.

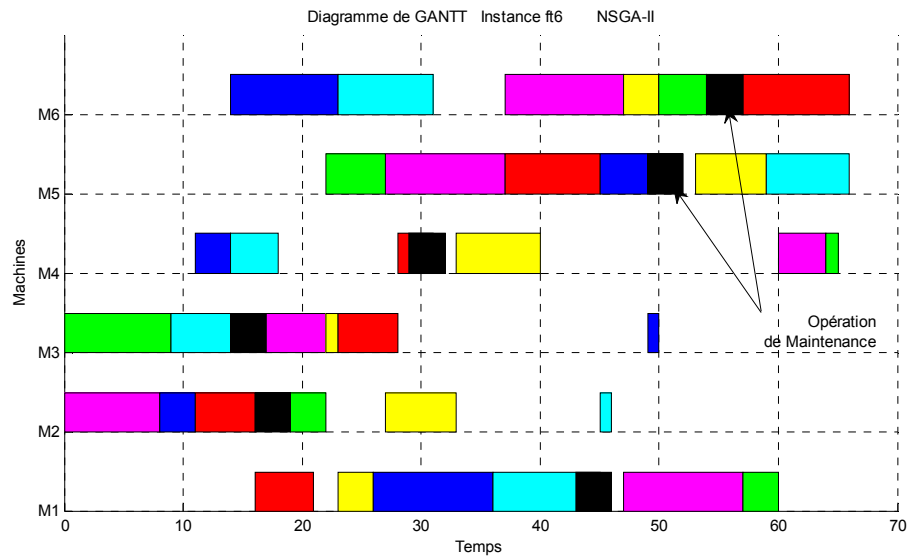


Figure (4.19) : Diagramme de GANTT de l'instance Ft06 sous contrainte d'indisponibilité.

Les résultats expérimentaux, montrent bien l'existence d'un antagonisme entre le Makespan d'une part et les coûts de production et de maintenance d'autre part, Figure (3.17). La variation du Makespan entraîne la variation dans le sens contraire des coûts de production et des coûts de maintenance, mais avec des proportionnalités différentes. Cet antagonisme, rend la tâche de l'utilisateur très compliquée quant au choix d'une solution adéquate à son problème. Étant donnée le nombre de solutions Pareto optimales offertes par l'algorithme NSGA-II, l'utilisateur peut être éclairé dans son choix par l'une de méthodes de tri (TOPSIS par exemple).

Le choix adéquat de la période de maintenance préventive systématique peut amortir le nombre d'intervention pour maintenance corrective, et le respect des dates d'intervention avec une flexibilité réfléchie, améliore considérablement la qualité de la productivité optimise le temps de séjour des produits dans l'atelier, les coûts de production et les coûts de maintenance des entreprises.

La prise en compte de flexibilité de la période de bon fonctionnement des machines permet de maintenir ces derniers en fonction de la période de travail réel (période productive de la machine), et non pas de toute la période même si la machine et non sollicitée. Le gain procuré par cette flexibilité est aussi bien au niveau de la production qu'au niveau de la maintenance.

## Conclusion

La persistance des entreprises dans des marchés fluctuant est conditionnée par l'optimisation des délais de mise sur le marché des produits et par la minimisation des coûts de la production et de la maintenance. La proposition des méthodes pour l'ordonnancement intégré des différentes fonctions de l'entreprise, particulièrement, la production et la maintenance, représente actuellement un atout possible pour lever ces défis.

Dans ce chapitre, une approche de génération des solutions Pareto optimales est présentée, pour résoudre le problème d'ordonnancement job-shop multicritère intégré des opérations de la production et de la maintenance. En effet, cet ordonnancement a permis d'avoir un meilleur compromis en minimisant simultanément deux critères de production (Makespan et coûts de production) et un critère de maintenance (coûts de maintenance).

Dans ce chapitre, nous avons pu atteindre deux objectifs. Le premier, consiste à l'introduction de la flexibilité de la période de disponibilité. Cette dernière est fonction de la charge de la machine. Le deuxième objectif repose sur la flexibilité de la date de début de l'indisponibilité, qui peut glisser dans une fenêtre de temps. Pour mettre en exergue ces deux cas d'études, nous avons proposé un algorithme génétique multicritère NSGA-II pour l'ordonnancement intégré de la production et de la maintenance. Cette approche est testée sur les 44 instances types. Elle a démontré son efficacité en générant des solutions avec des valeurs très proche de l'optimum, ceci en comparant les résultats obtenus avec les résultats d'autres recherches.

---

**CONCLUSION  
GÉNÉRALE ET  
PERSPECTIVES**

---

# Conclusion générale et perspectives

Les entreprises dans le contexte de mondialisation contemporain, doivent être de plus en plus compétitives. Évidemment, les moyens de production doivent s'exploiter d'une manière optimale. De notre temps, il est nécessaire de chercher d'autres leviers d'amélioration. D'une part, la fonction production doit : satisfaire la demande, maintenir voire accroître la capacité productive et tendre vers une meilleure qualité. L'affectation et l'ordonnancement des activités de production sur les équipements disponible, représentent donc un moyen d'action directe sur la productivité. D'autre part, les services maintenance interviennent pour maintenir ou remettre en état de bon fonctionnement les équipements indisponibles pour la production. L'affectation et l'ordonnancement des activités de maintenance représentent aussi, un moyen d'action directe sur la performance des entreprises.

Que ce soit dans le domaine de la gestion de production ou dans celui du pilotage d'un atelier de production, l'ordonnancement est un problème difficile qui recouvre souvent des enjeux économiques de première importance. Cet ordonnancement devient crucial lorsqu'il touche deux fonctions aussi importantes qu'antagonistes au sein de l'entreprise qui sont la fonction production et la fonction maintenance. Pour contribuer à ce problème conflictuel, nous avons proposé d'aborder dans ce travail le problème d'optimisation multicritères d'un ordonnancement intégrant les activités de la production et de la maintenance dans un atelier de type job shop.

Dans ce travail, nous avons abordé, les principales caractéristiques des problèmes d'ordonnancement de la production en générale et de manière plus particulière, les problèmes d'ordonnancement job shop. Nous nous sommes intéressés ensuite à la complexité des problèmes que peuvent être rencontrés, et ce pour situer à la fin le contexte de notre étude. Nous retenons que le service maintenance est fortement lié au service production, malheureusement, leurs objectifs sont antagonistes et demandent les mêmes ressources. Ainsi, l'ordonnancement des tâches de la maintenance et de la production séparés génère des conflits qui vont à l'encontre de la performance optimale des machines, notamment leurs disponibilités. Pour une meilleure cohérence entre les deux fonctions, le concept d'intégration s'avère indispensable. Dans la plupart des cas réels, on est confronté aux problèmes d'optimisation multicritères, caractérisés par la présence simultanée de plusieurs objectifs, souvent contradictoires. En effet, les approches de résolution multicritère consistaient pendant longtemps, principalement à la transformation des problèmes d'optimisation multicritères en un problème monocritère. Plusieurs approches ont été développées dans ce contexte. Ces approches nécessitent en réalité, des connaissances a priori sur le problème à traiter. En outre,

elles modifient la structure du problème combinatoire, qui perd ainsi, ses éventuelles propriétés remarquables. Or, dans les situations réelles, tel que le cas de notre problème, les décideurs ont généralement besoin de plusieurs alternatives. Ce qui est utile aussi pour une utilisation future, particulièrement quand la situation change et qu'une nouvelle solution est requise. Le décideur peut ainsi choisir une solution suivant la situation courante. Les approches Pareto présentent l'avantage de traiter les problèmes multicritères sans transformation, sans favoriser un objectif par rapport à un autre et peuvent générer des solutions supportées et non supportées en un passage unique de l'algorithme. Le choix d'une méthode de résolution à mettre en œuvre dépendra souvent de la complexité du problème. De part, leurs performances prouvées pour la résolution de problèmes complexes, l'algorithme NSGA-II (Non Dominated Sorting Genetic algorithm) représente à cet effet un outil bien adapté pour résoudre notre problème.

À travers la littérature consultée dédiée aux problèmes d'ordonnement Job Shop, en général et Job Shop avec possibilité d'indisponibilité des ressources en particulier, nous avons constaté que la majorité des travaux considèrent que la période de disponibilité est fixe et prédéterminée. De même la date de début d'intervention est supposée aussi fixe, à l'exception des travaux de [Azem et al. 2011] où ils ont considéré la flexibilité de la date d'intervention pour maintenance préventive, sans aucune pénalité d'avance et de retard. La plupart de ces travaux adoptent pour la résolution des problèmes des approches monocritère ou des approches qui transforme les problèmes multicritères en monocritère.

À la différence de la littérature consulté jusqu'à présent, nous avons considéré le cas de la double flexibilité c'est-à-dire, d'une part, les périodes de disponibilité qui varient en fonction de la charge des machines, d'autre part la date de début d'intervention peut glisser dans une fenêtre de temps avec pénalités d'avance et de retard. Les jobs de production sont eux aussi soumis à des pénalités d'avance et de retard pour assurer une production en juste à temps. Pour optimiser simultanément les critères considérés (coûts de production, coût de maintenance et le Makespan) nous avons adapté l'approche NSGA-II à notre problématique. En effet, NSGA-II a fait preuve dans le domaine aussi bien monocritère que multicritère.

Pour valider notre approche, nous avons présenté plusieurs résultats, issus de trois types de simulation : dans le premier nous avons considéré un seul critère (Makespan) avec disponibilité permanente des machines. Dans le second type, deux critères (Makespan et coûts de production) avec disponibilité permanente des machines ont été considérés. Dans le dernier, trois critères (Makespan et coûts de production et coûts de maintenance) avec possibilité d'indisponibilité des machines sont aussi considérés.

L'indisponibilité des machines est néfaste quant à la productivité et aux coûts de production. Sa prise en considération lors de l'ordonnancement avec intégration des tâches de la production et de la maintenance n'augmente que peu les coûts. Ces coûts restent certainement largement inférieurs au coût d'une panne inattendue.

Un problème d'ordonnancement intégrant les opérations de la production et de la maintenance est soumis à plusieurs facteurs de complexité, à savoir les facteurs liés à la typologie des systèmes de production (type d'atelier), les facteurs liés aux types de tâche de maintenance, les facteurs liés à l'horizon de planification et les facteurs liés aux nombres et compétences des ressources humaines de la maintenance.

### Perspectives

Les perspectives de recherche sont nombreuses :

- C'est vrai, une maintenance préventive systématique efficace peut contribuer à la minimisation du risque des pannes subites, mais malheureusement, elle ne les élimine pas totalement. L'occurrence d'une panne, nécessite une intervention corrective, le plus tôt possible pour remettre l'équipement en état de fonctionnement. Ainsi, pour éliminer les effets néfastes, de cette situation, nous proposons, d'étendre notre approche pour prendre en charge la maintenance corrective, ceci, en considérons des tâches de maintenance corrective qui s'intègrent d'une manière aléatoire dans l'ordonnancement des activités de la production et de maintenance systématique. De ce fait un autre critère d'optimisation des coûts de la maintenance corrective peut être pris en compte. D'après la nature des tâches de la maintenance corrective une approche floue logique peut être adoptée.
- Généralement, la longueur de la fenêtre de temps dont la date de début d'intervention préventive systématique peut glisser est prédéfini et souvent non optimale. Pour aider le décideur au moment du choix de cette dernière, nous envisageons de résoudre le problème en considérant comme critère, l'optimisation de longueur de la fenêtre de temps.
- La maintenance préventive systématique est supposée jusqu'ici capable de remettre l'équipement dans ses performances initiales. Or, cette hypothèse est loin de la réalité. Pour contribuer à cette problématique, nous proposons, de rendre l'ordonnancement plus dynamique, dans ce cas il faut insérer l'effet de la dégradation sur la durée de disponibilité des équipements.



---

**REFERANCE**  
**BOBLOGRAPHIQUE**

---

## Références Bibliographiques

### A

- ABBOU R., (2003). Contribution à la mise en œuvre d'une maintenance centralisée : conception et optimisation d'un atelier de maintenance. Thèse de Doctorat, Université Joseph Fourier, Grenoble, France.
- Adzakpa K. P., (2005). Maintenance des Systèmes Distribués : Méthodes d'aide à la Décision temps-réel. Thèse de doctorat en optimisation et sûreté des systèmes, de l'université de Technologie de Troyes, France.
- Aggoun A., et Beldiceanu N., (1993). Extending CHIP in Order to Solve Complex Scheduling Problems, *Journal of Mathematical and Computer Modeling*, Pergamon Press. V : 17 (7), pp : 57-73.
- Aghezzaf E.H., Jamali, M.A. et Ait-Kadi D., (2007). An integrated production and preventive maintenance planning model. *European Journal of Operational Research*, V : 18(1, 2), pp : 679-685.
- Aghezzaf, E.H. et Najid, M.N. (2008). Integrated production and preventive maintenance in production systems subject to random failures. *Information science*. V : 178, pp : 3382-3392.
- Al-Anzi F. S., Sotskov Y. N., Allahverdi A., et Andreev G. V., (2006). Using mixed graph coloring to minimize total completion time In job shop scheduling problem. *Applied mathematics and computation*. V : 182, pp : 1137-1148.
- Alberto P. S. and Marco T., (2012). Integrated maintenance and production planning: A model to include rescheduling costs. *Journal of Quality in Maintenance Engineering*. V : 18 (1), pp : 42-59.
- Allenson R., (1992). Genetic Algorithm with Gender for Multi-Function Optimisation, TR. EPCC-SS92-01, Edinburgh Parallel Computing Center, Edinburgh, Scotland.
- Ammar Oulamara, (2009). Contribution à l'étude des problèmes d'ordonnancement Flow shop avec contraintes supplémentaires : Complexité et méthodes de résolution. MÉMOIRE d'habilitation à Diriger des Recherches de l'Institut National Polytechnique de Lorraine, France.
- Audrey C., (2015). Bi-objective branch-and-cut algorithms applied to the binary knapsack problem. Thèse de Doctorat, en Informatique et applications, de l'Université de Nantes, France.

### B

- Bagchi T. P., (1999). Multiobjective Scheduling by Genetic Algorithms, Kluwer academic publishers.
- Baker K.R., (1974). Introduction to Sequencing and Scheduling. Wiley, New York.
- Balkaid F., Dahane M., (2014). Optimisation des services de maintenance et de production sous des contraintes de ressources non-renouvelables : une approche basée sur les Algorithmes Génétiques. 10ème Conférence Francophone de Modélisation, Optimisation et Simulation- MOSIM'14, Nancy –France.
- Baptiste P., Le Pape C., et Péridy L., (1998). Global constraints for partial CSPs : a case study of resource and due-date constraints. International conference on Principales and Practice of Constraint Programming, Pisa, Italy.
- Barichard V., (2003). Approches hybrides pour les problèmes multiobjectifs. Thèse de doctorat en Informatique de l'école Doctorale d'Angers, France.
- Beasley D., Bull D.R., et Martin R.R., (1993). A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, V : 1 (2), pp : 101-125.
- Bellone, J. Chamard, A., et Pradelles C., (1992). PLANE -An Evolutive Planning System for Aircraft Production. First International Conference on the Practical Application of Prolog, London.
- Benameur L. (2010). Contribution à l'optimisation complexe par des techniques de swarm intelligence. Thèse de doctorat en Informatique et Télécommunications de l'université Mohammed V – Agdal Rabat, Maroc.
- Benbouzid S. F., (2005). Contribution à l'étude de la performance et de la robustesse des ordonnancements conjoints Production/Maintenance : Cas du Flow shop. Thèse de doctorat, en Automatique et Informatique. UFR des Sciences et Techniques de l'Université de Franche-Comté, France.
- Berro A., (2001). Optimisation multiobjectif et stratégies d'évolution en environnement dynamique. Thèse de doctorat en informatique de l'université des sciences sociales Toulouse I, France.
- Billaut J.-C., et Moukrim A., (2005). Introduction à la flexibilité et à la robustesse en ordonnancement dans Flexibilité et robustesse en ordonnancement. Sciences publications Lavoisier Hermes. Paris.

- Bisdorff, R Laurent S., Pichon E., (1995). Knowledge Engineering with CHIP - Application to a Production Scheduling Problem in the Wire-Drawing Industry PAP'95, Paris.
- Blazewicz J., Ecker K.H, Pesch E., Schmidt G., and Weglarz J., (1996). Scheduling Computer and Manufacturing Processes. Springer, Berlin.
- Blazewicz J., Ecker K.H, Pesch E., Schmidt G., and Weglarz J., (2007). Handbook on scheduling from Theory to Applications. International Handbooks on Information systems.
- Blickle T. (1996). Evolving compact solutions in genetic programming : A case study. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, Parallel Problem Solving From Nature IV. Proceedings of the International Conference on Evolutionary Computation., pp : 564-573, Heidelberg, Germany, Springer-Verlag, LNCS 1141.
- Blondel F., (2004). Gestion de la production. 3ème édition, DUNOD, Paris, France.
- Boizumault P., Delon Y., et Peridy L., (1994). Planning Exams Using Constraint Logic Programming, 2nd Conf Practical Applications of Prolog, London.
- Boucly F. et Ogus A., (1987). Le management de la maintenance. AFNOR, Paris, France.
- Brandolese M., Fransi M. et Pozzetti A., (1996). Production and maintenance integrated planning. International Journal of Production Research, V : 34 (7), pp : 2059-2075, 1996.
- Brits R., Engelbrecht A., et Van den Bergh F., (2002-1). Solving systems of unconstrained equations using particle swarm optimization. Proceedings of the IEEE Conf. on Systems, Man, and Cybernetics, Tunisa, V : 3, pp : 6.
- Brits R., Engelbrecht A., et Van den Bergh F., (2002-2). A niching particle swarm optimizer. Wang, L., Tan, K.C., Furuhashi, T., Kim, J.H., Yao, X., eds., Proceedings of the 4th Asia-Pacific Conf. on Simulated Evolution and Learning, V : 2, pp : 692-696, Singapore.
- Brucker P., (1998). Scheduling Algorithms. Springer-Verlag, Berlin Heidelberg.
- Cai X., Wu X., et Zhou X., (2014). Optimal Stochastic Scheduling. Springer US. V : 207. ISBN: 978-1-4899-7404-4.
- C**
- Carlier, J. et Chrétienne P., (1988). Problèmes d'ordonnancement, Modélisation, Complexité, Algorithmes. Edition Masson, Paris, France,
- Carraway R.L., Morin T.L., et Moskowitz H., (1990). Generalized dynamic programming for multicriteria optimization. European Journal of Operational Research, V : 44, pp : 95-104.
- Charles A.S., (2000). Modélisation des défaillances des équipements d'une unité de production de composants électroniques. Optimisation des stratégies de maintenance. Thèse de Doctorat, Institut National Polytechnique de Toulouse, France.
- Chrétienne P., Coffman E.G., Liu Z., et Lenstra J.K., (1995). Scheduling theory and its applications. Wiley, New York, Etats unis.
- Chu C., et Proth J.M., (1996). L'ordonnancement et ses applications. Masson, Paris, France.
- Mortan T.E. et Pintico D. (1993). Heuristics scheduling systems. Wiley, New York, Etats-Unis.
- Clarisse D. F., (2005) .Optimisation Combinatoire Multi-Objectif : Apport des méthodes coopératives et contribution à l'extraction de connaissances. Thèse d'Habilitation à Diriger des Recherches de l'U.S.T.L. Université des Sciences et Technologies de Lille, France.
- Coello C.A., (1998). Using a min-max method to solve multiobjective optimization problems with genetic algorithms. In IBERAMIA'98, LNCS, Springer Verlag. V : 1993, pp : 303-314.
- Coffman E. G Jr., (1976). Computer and job shop Scheduling Theory. John. Wiley and sons, New York.
- Collette Y. et Siarry P., (2002). Optimisation multiobjectif. Éditions Eyrolles, ISBN 2-212-11168-1, France.
- Combaz J. (2006). Conception de Systèmes Adaptatifs Sûrs et Optimaux. Thèse de doctorat en Mathématique Informatique de l'université Joseph Fourier, France.
- Conway R.W, Maxwell W.L., et Miller L.W., (1967). Theory of Scheduling. Addison Wesley, Reading, Mass., USA.
- Cook S., (1985). A Taxonomy of Problems with Fast Parallel Algorithms. Information and Control, V : 64, pp : 2-22.

Corne D. W., et al, (2001). PESA II : Region-based Selection in Evolutionary Multiobjective Optimization, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001). pp : 283-290, San Francisco, California.

Coudert T., (2000). Apport des systèmes multi-agents pour la négociation en ordonnancement : application aux fonctions production et maintenance. Thèse de doctorat soutenue à l'ENI de Tarbes, France.

Creemers T., Giralt L.R., Riera J, Ferrarons C., Rocca J., Corbella X., (1995). Constrained-Based Maintenance Scheduling on an Electric Power-Distribution Network, PAP95, Paris.

CSP2SAT, (2017), JSS benchmark results. <http://bach.istc.kobe-u.ac.jp/csp2sat/jss/>.

## D

De Jong, K. A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. thesis, University of Michigan.

Deb K., (2001). Multi-objective optimization using evolutionary algorithms. John Wiley.

Deb K., Pratap A., et Meyarivan T., (2001). Constrained test problems for multi-objective evolutionary optimization. In Proceedings of Evolutionary Multi-Criterion Optimization, pp : 284-298.

De-Castro L. N., et Timmis J., (2002). An artificial Immune Network for multimodal function optimization. The IEEE World Congress on Computational Intelligence.

Deloux E., Castanier B., and Bérenguer C., (2007). Optimisation de la politique de maintenance pour un système à dégradation graduelle stressé, 7ème Congrès international pluridisciplinaire Qualité de Sécurité de Fonctionnement. V : 1, pp: 142-149, Tanger, Maroc.

Diebolt D., (2007). Diagnostic de la maintenance - Mesure de performance et interprétation. Techniques de l'Ingénieur, Référence MT9130.

Doumeings G., (1984). Méthode GRAI : méthode de conception des systèmes en productique. Thèse d'état soutenue à l'université de Bordeaux-I, France.

## E

Edgeworth F.Y., (1881). Mathematical physics. P. Heagan, Londres, Angleterre.

Efaga E.D., (2004). Analyse des données du retour d'expérience pour l'organisation de la maintenance des équipements de production des PME/PMI dans le cadre de la MBF (maintenance Basée sur la Fiabilité). Thèse de l'Université Louis Pasteur de Strasbourg, France.

Egèse R.W., (1990). Simulated annealing: A tool for operational research, European Journal of Operational Research. V :46 (3): pp : 271-281.

Ehrgott M., (2000). Multicriteria optimization. In Lecture Notes in Economics and Mathematical Systems, Springer. V : 491.

Ehrgott M., (2005). Multicriteria optimization, volume 491 of Lecture Notes in Economics and Mathematical Systems. Springer, 2nd ed. edition.

Ehrgott M., et Gandibleux X., (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum. V : 22, pp : 425-460.

El Imrani A. A., Bouroumi A., Zine El Abidine H., Limouri M., et Essaïd A., (1999). A Fuzzy Clustering-based Niching Approach to Multimodal Function Optimization. Journal Cognitive Systems research, V : 1 (2), pp : 119-133.

El Imrani A.A., (2000). Conception d'un Algorithme Génétique Coévolutif. Application à l'optimisation des Histoires Thermiques. Thèse de Doctorat d'Etat. Faculté des Sciences, Rabat.

Emmons H., Vairaktarakis G., (2013). Flow Shop Scheduling : Theoretical Results, Algorithms, and Applications. Springer US. V : 182. ISBN: 978-1-4614-5151-8.

Eshelman L. J. et Schaffer J. D., (1991). Preventing premature convergence in genetic algorithms by preventing incest. In R. K. Belew and L. B. Booker, editors, Fourth Int. Conf. on Genetic Algorithms ICGA'4, Morgan Kaufmann Pub. pp : 115-122, San Mateo, California.

Esquirol P. et Lopez P, (1999). L'ordonnancement. Économica, ISBN : 2717837981. Paris France.

## F

Fleurquin G, Riane F, Roland B, Dehombreux P, et Hou G, (2006). Optimain : Plate forme d'évaluation de stratégie de maintenance des systèmes complexes. MOSIM'06.

- Fonseca C. M. et Fleming P. J., (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*. V : 3 (1), pp : 1-16.
- Fonseca C., (1995), Multiobjective genetic algorithms with applications to control engineering problems. PhD thesis, University of Sheffield.
- Fontanili F., (1999). Intégration d'outils de simulation et d'optimisation pour le pilotage d'une ligne d'assemblage multi-produit à transfert asynchrone, thèse de doctorat, Université Paris XIII, France.
- Fourman, (1985). Compaction of Symbolic Layout using Genetic Algorithms. In *Genetic Algorithms and their Applications : Proceedings of the First International Conference on Genetic Algorithm*, pp : 141-153.
- Framinan J. M., Leisten R., et García R. R., (2014). *Manufacturing Scheduling Systems : An Integrated View on Models, Methods and Tools*. Springer London. ISBN: 978-1-4471-6271-1.
- Frédéric Planchet, (2015). Introduction. Support de cours 2015-2016, Version 3.8.
- French S., (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Horwood, Chichester.
- Friesz et al. (1993). The multiobjective equilibrium network design problem revisited : a simulated annealing approach. *European Journal of Operational Research*. V : 65, pp : 44-57.
- Fujita K., Hirokawa N., Akagi S., Kimatura S. et Yokohata H., (1998). Multi-objective optimal design of automotive engine using genetic algorithm. In *Design Engineering Technical Conferences DETC'98*, pp : 1-11, Atlanta, Georgia.

## G

- Garey M.R., Johnson D.S. et Sethi R., (1976). The complexity of flow shop and job shop scheduling. *Mathematics of Operations Research*. V : 1, pp: 117–129.
- Giard V., (1988). *Gestion de la Production*. 2ème édition, Économisa, Paris, France.
- Giard V., (1991). Une comptabilité de gestion en crise. ECOSIP. Économisa, pp : 135-149, Paris, France.
- Glover F., (1986). Future paths for integer programming and links to artificial intelligence, *Comput. And Ops. Res.* V : 13 (5), pp : 533-549.
- Glover F., (1990). Artificial intelligence, heuristic frameworks and tabu search, *Managerial and decision economics*, V : 11, pp : 365-375.
- Glover F., et Laguna M., (1997). *Tabu search*, Kluwer Academic Publishers.
- Goldberg D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, New York.
- Goldberg D. E., et Richardson J., (1987). Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms*, pp : 41-49.
- Goldberg D. E., et Wang L., (1997). Adaptive niching via coevolutionary sharing. Quagliarella et al.(Eds.). *Genetic Algorithms in Engineering and Computer Science* (John Wiley and Sons, Ltd), pp : 21-38.
- Goldberg D.E., (1989). Genetic algorithms for search, optimization, and machine learning. In : Addison-Wesley, MA : (ed), Reading.

## H

- Haimes Y., Lasdon L., et Wismer D., (1971). On a bicriterion formulation for the problems of integrated system identification and system optimization. *IEEE Transaction on Systems, Man and Cybernetics*. V : 1, pp : 296–297.
- Chankong V. et Haimes Y.(1983). *Multiobjective decision making : Theory and methodology*. Elsevier Science Publishing Co.
- Hajela P. et Lin C. Y., (1992). Genetic search strategies in multicriterion optimal design. *Structural Optimization*. V : 4, pp : 99-107.
- Halimi Djamel, (2014). Contribution à l'amélioration de la maintenance préventive des machines dynamiqués dans l'industrie des hydrocarbures. Thèse de doctorat en Génie Mécanique, Université M'Hamed Bougara-Boumerdes, Algérie.
- Hentous H, (1999). Contribution au pilotage des systèmes de production de type job shop. Thèse de doctorat, INSA Lyon, France.

Hertz A., Jaumard B., Ribeiro C.C., et Formosinho Filho W.P., (1994). A multi-criteria tabu search approach to cell formation problems in group technology with multiple objectives. *RAIRO Operations Research*. V : 28 (3), pp : 303–328.

Hoël C., (2011). Optimisation sans dérivées sous contraintes : deux applications industrielles en ingénierie de réservoir et en calibration des moteurs. Thèse de doctorat en Automatique, Traitement du Signal et des Images, de l'Université de Nice-Sophia Antipolis Mention, France.

Hopp W.J., et Kuo Y.L., (1998). An optimal structured policy for maintenance of partially observable aircraft engine components. *Naval Research Logistics*. V : 45 (4), pp : 335-352.

## I

Irisarri F. X. (2009). Stratégies de calcul pour l'optimisation multiobjectif des structures composites. Thèse de doctorat, Université Toulouse III Paul Sabatier, France.

Ishibuchi H. et Murata T. (1996). Multi-objective genetic local search algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'96)*. pp : 119-124.

## J

Jacquin S., (2015). Hybridation des méthodes et de la programmation dynamique pour les problèmes d'optimisation mono et multi-objectif : application à la production d'énergie. Thèse de doctorat en informatique, laboratoire CRISTAL, Université de Lille 1, France.

Jain A.S., et Meeran S., (1999). Deterministic job-shop scheduling: Past, present and future » *European Journal of Operational Research*. V: 113, pp: 390–434.

Javel G., (2004). *Organisation et Gestion de la Production*, 3ème édition, DUNOD, Paris France.

Jeffrey W H., (2006). *Handbook of Production Scheduling*. SBN: 978-0-387-33115-7. Springer New York.

Jeong-hwa AN, (2011). Le choix d'un système de transport durable : analyse comparative des systèmes de transport guidé de surface. Thèse de doctorat en Transport de l'université Paris-Est, France.

Jérémi R., (2003). Conception de systèmes hétérogènes en Génie Électrique par optimisation évolutionnaire multicritère. Doctorat en Génie Électrique, de L'institut Nationale Polytechnique de Toulouse, France.

Jozefowicz N., (2013). Optimisation combinatoire multi-objectif : des méthodes aux problèmes, de la Terre à (presque) la Lune. Habilitation à diriger les recherches en informatique. Institut National Polytechnique de Toulouse, France.

## K

Kacem I. (2003), Ordonnancement multicritère des job-Shop flexibles : formulation, bornes inférieures et approche évolutionniste coopérative. Thèse de Doctorat, Université des Sciences et Techniques de Lille1.

Kennedy J., (2000). Stereotyping : Improving particle swarm performance with cluster analysis. *Proceedings of the IEEE Congress on Evol. Comput.* (San Diego, California, U.S.A.), pp : 1507-1512.

Knowles J. D. et Corne D. W., (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*. V : 8 (2), pp : 149–172.

Knowles J. et Corne D., (1999). Pareto archived evolution strategy : A new baseline algorithm for Pareto multiobjective optimisation. In *1999 Congress on Evolutionary Computaton (CEC'99)*. V : 1, pp : 98–105.

Kołodziej J., (2012). *Evolutionary Hierarchical Multi-Criteria Metaheuristics for Scheduling in Large-Scale Grid Systems*. Springer Berlin Heidelberg. V : 419. ISBN: 978-3-642-28970-5.

## L

Lavina Yves, (1992). *Audit de la maintenance*. Les Éditions d'organisation, Paris, France.

Lei D., (2013). Multi-objective artificial bee colony for interval job shop scheduling with flexible maintenance". *Int J Adv Manuf Technol*. V : 66, pp: 1835–1843.

Lenstra J.K. et A.H.G. Rinnooy Khan, (1977). Complexity of machine scheduling problems. *Ann. Discrete Math*. Vol: 1, pp: 343-362.

Letouzey A., (2001). Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs », Thèse de Doctorat, Institut National Polytechnique de Toulouse, France.

Li J.Q., et Pan Q.K., (2013). Chemical-reaction optimization for solving fuzzy job-shop scheduling problem with flexible maintenance activities. *Int. J. Production Economics*. V: 145, pp: 4-17.



Li J.Q., Pan Q.K. et Tasgetiren M. F., (2014). A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities. *Applied Mathematical Modelling*. V: 38, pp: 1111–1132.

Li X., (2004). Adaptively choosing neighborhood bests using species in swarm optimizer for multimodal function optimization. *GECCO*. pp : 105-116.

Lis J., et Eiben A.E., (1996). A Multi-Sexual Genetic Algorithm for Multiobjective Optimization, In T.Fukuda and T. Furuhashi ed., *Proceedings of the 1996 International Conference on Evolutionary Computation*, Nagoya, Japan. pp.59-64.

Lopez P., et Roubellat F., (2001). *Ordonnancement de la Production*. Hermes Science, France.

Loughlin D. H. et Ranjithan S., (1997). The neighborhood constraint method : A genetic algorithm-based multiobjective optimization technique. In T. Back, editor, *Seventh Int. Conf. on Genetic Algorithms ICGA'97*, Morgan Kaufmann. pp : 666-6773, San Mateo, California.

## M

Ma Y., Chu C.B., et Zuo C.R., (2010). A survey of scheduling with deterministic machine availability constraints. *Computers and Industrial Engineering*. V: 58, pp: 199–211.

Mahfoud S.W., (1994). Crossover Interactions Among niches. *Proceedings of the 1st IEEE Conference on Evolutionary Computation*. pp : 188-193.

Mahfoud S.W., (1995). Niching methods for genetic algorithms. Doctoral dissertation/Technical report No. 95001, University of Illinois at Urbana Champaign, Illinois Genetic Algorithm Laboratory.

Mammeri M., (2013). Une approche d'aide multicritère à la décision pour l'évaluation du confort dans les trains : construction d'un modèle d'évaluation. Thèse de doctorat en Informatique, de l'École Doctorale de Dauphine (ED543), Université Paris-Dauphine, France.

Mandow L. et Millan E., (1996). Goal programming and heuristic search. In R. Caballero, F. Ruiz, and R. Steuer, editors, *Second Int. Conf. on Multi-Objective Programming and Goal Programming MOPGP'96*, Springer-Verlag. pp : 48-56, Torremolinos, Spain.

Manne A.S., (1960). On the Job-Shop Scheduling Problem, *Operations Research*. V : 8 (2): pp.219-223.

Mardle S., Pascoes S., et Tamiz M., (2000). An investigation of genetic algorithm for the optimization of multi-objective fisheries bioeconomic models. *International Transaction of Operation research*. V : 7, pp : 33–49.

Mati Y., (2010). Minimizing the makespan in the non-preemptive job-shop scheduling with limited machine availability. *Computers and Industrial Engineering*. V: 59, pp: 537-543.

Mauergauz Y., (2013). *Advanced Planning and Scheduling in Manufacturing and Supply Chains*. Springer Berlin Heidelberg. V : 182, ISBN: 978-3-319-27521-5.

Mavrotas G., et Diakoulaki D., (1998). A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*. Order. V : 107, pp : 530-541.

Mavrotas G., et Diakoulaki D., (2005). Multi-criteria branch and bound: A vector maximization algorithm for mixed 0-1 multiple objective linear programming. *Applied Mathematics and Computation*. Order. V : 171, pp : 53-71.

Miettinen K., (1999). *Nonlinear multiobjective optimization*. Kluwer Academic Publishers.

Miller B.L., et Shaw M.J., (1996). Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. *IlligAL Report no : 95010*.

Moghaddam K.S. (2013). Multi-objective maintenance and replacement scheduling in a manufacturing using goal programming. *Int. J. Production Economics*. V: 146, pp: 704-716.

Monchy F., (2000). *Maintenance : Méthodes et Organisations*. Edition DUNOD.

Monchy F., et Monchy J. P, V., (2003). *La maintenance, méthode et organisation*. 3ème édition Dunod.

Moubray J., (2000) *Reliability-Centered Maintenance*. industrial press, Second Edition, ISBN-13: 978-0831131463. New York.

Murata T., et Ishibuchi H., (1998). A multi-objectives genetic local search algorithm and its application flow-shop scheduling. *IEEE Transaction System*. V : 28 (3), pp : 392–403.

## O

Ogus A. (1988). *Maintenance*. Volume A8450. Techniques de l'Ingénieur, Paris, France.

- Olivier L. D. W. (2004). Multiobjective Optimization: History and Promise. CJK-OSM3, Kanazawa.
- Ombuki B.M., et Ventresca M., (2004). Local search genetic algorithms for the job shop scheduling problem. *Applied Intelligence*, V: 21 (1), pp: 99-109.

**P**

- Paquete L. et Stützle T., (2006). A study of stochastic local search algorithms for the biobjective QAP with correlated flow matrices. *European Journal of Operational Research*. V : 169, pp : 943–959.
- Pareto V., (1896). *Cours d'économie politique*. Rouge, Lausanne, Suisse.
- Parks G.T. et Miller I., (1998). Selective breeding in a multiobjective genetic algorithm. In *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Springer. pp : 250-259.
- Petrowsky A., (1996). A clearing procedure as a niching method for Genetic Algorithms. *Proceedings of IEEE International Conference of Evolutionary Computation*, (Nagoya-Japan). pp : 798-803.
- Pinedo M. L., (2012). *Scheduling : Theory, Algorithms, and Systems*. Springer US. ISBN: 978-1-4614-1986-0.
- Pinedo M. L., (2016). *Scheduling : Theory, Algorithms, and Systems*. Springer International Publishing. ISBN: 978-3-319-26578-0.
- Pinedo M., (1995). *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, Etats-Unis.
- Pinedo M., (2008). *Scheduling : Theory, algorithms and systems*. (2nd edition), Prentice Hall.
- Pinson E. (1995), The job shop scheduling problem: A concise survey and some recent developments. In: Chrétienne P, Coffman EG, Lenstra JK, Liu Z (Eds.). *Scheduling Theory and Its Applications*. New York: Wiley;: 277-293.

**R**

- Rémi Laronde (2011) .Fiabilité et durabilité d'un système complexe dédié aux énergies renouvelables : Application à un système photovoltaïque. Thèse de doctorat, en Sciences de l'ingénieur. École doctorale sciences et technologies de l'information et mathématiques. Université Angers.
- Reza H., Saidy D. et TaghiTaghavi-F. M., (2008). Study of Scheduling Problems with Machine Availability Constraint. *Journal of Industrial and Systems Engineering*. V: 1 (4), pp: 360-383.
- Rinnooy K. A.H.G., (1976). *Machine Scheduling Problems: Classification, Complexity and Computations*. Martinus Nijhoff, The Hague.
- Roboam M., (1988). Modèles de référence et intégration des méthodes d'analyse pour la conception des systèmes de production. Thèse de doctorat soutenue à l'université de Bordeaux, France.
- Rodammer F.A. et Preston W. K., (1999). A recent survey of production scheduling. *IEEE Transaction on Systems, Man and Cybernetics*. pp. 6-18.
- Rosa A., (2007). Contribution à la mise en œuvre d'une maintenance centralisée : Conception et Optimisation d'un Atelier de Maintenance. Thèse de Doctorat DE L'UJF, Spécialité, automatique productique. Université Joseph Fourier Grenoble I, France.
- Roudenko O., (2004). Application des algorithmes évolutionnaires aux problèmes d'optimisation multi-objectif avec contraintes. Thèse en mathématiques appliquées de l'école Polytechnique.
- Roy B. et Bouysson D, (1993). Aide multi-critères à la décision : Méthodes et cas. Collection Gestion Série : Production et technologie quantitatives appliquées à la gestion. Edition Economica, Paris.

**S**

- Sareni B., et Krähenbühl L., (1998). Fitness Sharing and Niching Methods Revisited. *IEEE Transactions on Evolutionary Computation*, V : 2 (3), pp : 97-106.
- Schaffer D., (1985). Multiple Objective Optimisation with Vector Evaluated Genetic Algorithm, In *genetic Algorithm and their Applications : Proceedings of the First International Conference on Genetic Algorithm*. Pp : 93-100.
- Schmidt, G., (2000). Scheduling with limited machine availability. *European Journal of Operational Research*. V: 121, pp: 1–15.
- Schoeman I.L., et Engelbrecht A., (2005). A parallel vector-based particle swarm optimizer. *International Conf. on Artificial Neural Networks and Genetic Algorithms (ICANNGA)*, Portugal.
- Serafini P. (1992). Simulated annealing for multiple objective optimization problems. *Proceedings of the 10th International Conference on Multiple Criteria Decision*, Springer. pp : 87-96.



- Siarry P., (1999). Optimisation et classification de données, Cours du D.E.A G.B.M. de l'université de Paris 12.
- Simeu-A. Z., (1998). Sur la sûreté de fonctionnement des systèmes de production. Mémoire d'habilitation à diriger des recherches, Institut National Polytechnique de Grenoble, France.
- Smairi N., (2013). Contribution à l'Optimisation par Essaim Particulaire : adaptation de TRIBES à l'optimisation multiobjectif. Thèse de doctorat en Informatique, de l'école nationale des sciences de l'informatique, université de Manouba.
- Sourd F. et Spanjaard O., (2008). A Multiobjective Branch-and-Bound Framework: Application to the Biobjective Spanning Tree Problem. *Inform Journal on Computing*. V : 20 (3), pp : 472-484.
- Souris J.P., (1990). La maintenance source de profits. ISBN : 2708111361, Éditions d'organisation.
- Srinivas N., et Deb K., (1994). Multiobjective optimization using nondominated sorting in genetics algorithms. *Evolutionary Computation*. V : 2 (3), pp : 221-248.
- Srinivas S., (1995). A polynomial algorithm for computing the optimal repair strategy in a system with independent component failures. In *Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence*, Montreal, Canada.
- Stewart B.S. et White C.C., (1991). Multiobjective A\*. *Journal of the ACM*. V : 38 (4), pp : 775-814.

## T

- Talbi E., Rahoual M., Mabed M., et Dhaenens C., (2001). A hybrid evolutionary approach for multicriteria optimization problems : Applications to the flow shop. In E. Zitzler et al., editors, *Evolutionary Multi-Criterion Optimization*, of *Lecture Notes in Computer Science*. V : 1993, pp : 416-428.
- Tanaev V.S., Gordon V.S., Shafransky Y.M., (1994). *Scheduling theory. Single-stage systems*. Kluwer Academic Publishers. Dordrecht / Boston / London.

## U

- Ulungu E.L. et Teghem J., (1995). The two phases method : An efficient procedure to solve biobjective combinatorial optimization problems. *Foundation of computing and decision science*. V : 20, pp : 149-156.
- Ulungu E.L., et Teghem J., (1994). Multi-objective combinatorial optimization: a survey. *Journal of Multi-Criteria Decision Analysis*. V : 3, pp : 83-04.
- Ulungu, E., (1993). Optimisation combinatoire multicritère : détermination de l'ensemble des solutions efficaces et méthodes interactives. Thesis PhD, Université de MonsHainaut.

## V

- Vacher J Ph., (2000). Un système adaptatif par agents avec utilisation des algorithmes génétiques multi-objectifs : Application à l'ordonnancement d'atelier de type job-shop  $N \times M$ . Thèse de Doctorat, Université du Havre.
- Valdez-Flores C., Feldman R., (1989). A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *International Journal of Naval Research Logistics*. V : 36 (4), pp : 419-446.
- Vanhoucke M., (2013). *Project Management with Dynamic Scheduling : Springer Berlin Heidelberg. Baseline Scheduling, Risk Analysis and Project Control*. ISBN: 978-3-642-40437-5.
- Vassiliadis C.G, Arvela J., Pistikopoulos E.N., et Papageorgiou L.G., (2000). Planning and maintenance optimization for multipurpose plants. *Computer Aided Chemical Engineering*. V 8 (1), pp : 1105-1110.
- Vrignat P., Manuel A., Florent D., Sébastien A., Mohamed S., et Frédéric K., (2012). Génération d'indicateurs de maintenance par une approche semi-paramétrique et par une approche markovienne. *Revue Sciences et Maintenance, Association française des ingénieurs et responsables de maintenance (AFIM)*. pp : 1-28.

## Z

- Zaghdoud R., (2015). Hybridation d'algorithme génétique pour les problèmes des véhicules intelligents autonomes : applications aux infrastructures portuaires de moyenne taille. Thèse de doctorat en Automatique, Génie Informatique, Traitement du Signal et Images, de l'école centrale de Lille, France.
- Zhang J., Huang D. S., Lok T., et Lyu M. R., (2006). A novel adaptive sequential niche technique for multimodal function optimization. *Neurocomputing*. V : 69, pp : 2396-2401.
- Ziaee M., (2014). An efficient heuristic algorithm for flexible job shop scheduling with maintenance constraints. *Applied Mathematics and Sciences : An International Journal (MathSJ)*. V: 1.

- Zitzler E. et Thiele L., (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*. V : 3, pp : 257-271.
- Zouakia R., Bouami D., et Tkiouat M., (1999). Industrial systems maintenance modelling using Petri nets. *Reliability Engineering and System Safety*. V : 65 (2), pp : 119-124.
- Zribi N., Kamel A., et Borne P., (2008). Minimizing the makespan for the MPM job-shop with availability constraints. *International Journal of Production Economics*. V: 112 (1), pp: 151-160.
- Zwingelstein G., (1995). Diagnostic des défaillances, théorie et pratique pour les systèmes industriels. Ed. Hermes.