



HAL
open science

Ordonnancement d'ateliers de traitements de surfaces pour une production mono-robot/multi-produits : Résolution et étude de la robustesse

Imen Mhedhbi

► **To cite this version:**

Imen Mhedhbi. Ordonnancement d'ateliers de traitements de surfaces pour une production mono-robot/multi-produits : Résolution et étude de la robustesse. Autre. Ecole Centrale de Lille; École nationale d'ingénieurs de Tunis (Tunisie), 2011. Français. NNT : 2011ECLI0004 . tel-00992026

HAL Id: tel-00992026

<https://theses.hal.science/tel-00992026>

Submitted on 16 May 2014

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

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

N° d'ordre : 151

ECOLE CENTRALE DE LILLE
UNIVERSITE DE TUNIS EL MANAR
ECOLE NATIONALE D'INGENIEURS DE TUNIS

THESE

présentée en vue d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique, génie informatique, traitement du signal et images

par

Imen MHEDHBI BRINIS

DOCTORAT DELIVRE SIMULTANEMENT PAR L'ECOLE CENTRALE DE LILLE
ET L'ECOLE NATIONALE D'INGENIEURS DE TUNIS
DANS LE CADRE D'UNE COTUTELLE INTERNATIONALE DE THESE

Titre de la thèse :

**Ordonnancement d'ateliers de traitements de surfaces
pour une production mono-robot/multi-produits.**

Résolution et étude de la robustesse

soutenue le 11 avril 2011 devant le Jury d'Examen :

Président de jury	Ouajdi KORBAA	Professeur à l'ISITCom Sousse
Rapporteur	Nidhal REZG	Professeur, Directeur du LGIPM
Rapporteur	Moncef TAGINA	Professeur au LI3 / ENSI
Directeur de thèse	Etienne CRAYE	Professeur, Directeur de l'EC Lille
Directeur de thèse	Mohamed BENREJEB	Professeur à l'ENIT
Co-encadrant	Hervé CAMUS	Maître de Conférences à l'EC Lille

Thèse préparée dans les Laboratoires LAGIS (EC Lille) et LARA (ENIT)

Ecole Doctorale SPI 072 (EC Lille)
PRES Université Lille Nord-de-France

A mes parents

A ma sœur et mes frères

A mon cher mari

AVANT PROPOS

Le travail faisant l'objet du présent mémoire a été effectué au sein du Laboratoire de Recherche en Automatique (UR-LARA) de l'Ecole Nationale d'Ingénieurs de Tunis (ENIT) et du Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de l'Ecole Centrale de Lille (EC-Lille).

Nous tenons à exprimer notre vive gratitude à Monsieur Ouajdi KORBAA, à l'ISIT COM de l'Université de Sousse, pour nous avoir fait le grand honneur d'accepter de présider le Jury d'Examen. Qu'il trouve ici l'expression de notre profond respect.

C'est un agréable devoir pour nous d'exprimer notre très vive reconnaissance à Monsieur Etienne CRAYE, Directeur de l'EC-Lille, à Monsieur Hervé CAMUS, Maître de Conférences à l'EC-Lille et à Monsieur le Professeur Mohamed BENREJEB, Directeur de l'Unité de Recherche UR-LARA Automatique à l'ENIT, pour nous avoir guidé durant toute l'élaboration de ce mémoire avec le sérieux et la compétence qui les caractérisent. Qu'ils trouvent ici le témoignage de notre très profonde gratitude.

Nous tenons à remercier vivement Monsieur Moncef TAGINA, Professeur de l'ENSI, d'avoir accepté de rapporter notre travail. Qu'il soit grandement remercié.

Notre profonde gratitude à Monsieur Nidhal REZG, Professeur de l'Université Paul Verlaine de Metz, pour avoir bien voulu rapporter sur nos travaux de Thèse. Nous lui adressons nos sincères remerciements.

Nous tenons, enfin, à remercier tous les chercheurs du l'Unité de Recherche Automatique de l'ENIT et du Laboratoire d'Automatique, Génie Informatique et Signal de l'EC-Lille et toute personne qui, par leur aide, par leur amicale présence et par leur sympathie, ont rendu ce travail agréable. Nous leur exprimons ici toute notre gratitude.

SOMMAIRE

AVANT PROPOS.....	3
SOMMAIRE	4
LISTE DES TABLEAUX	9
LISTE DES FIGURES.....	11
INTRODUCTION GENERALE	13
<i>I</i> Chapitre I : Problème d’ordonnement des lignes de traitement de surfaces - Contraintes & Méthodes de résolution	21
I.1. Introduction	21
I.2. Spécificités des lignes de traitement de surfaces	22
I.2.1. Description physique de l’atelier étudié.....	22
I.2.1.1. Les cuves.....	23
I.2.1.2. Les robots.....	23
I.2.2. Le pilotage des lignes de traitement de surfaces	24
I.2.2.1. Problèmes d’ordonnement.....	24
• Caractérisation.....	24
• Les contraintes.....	25
• Les critères d’optimisation	27
I.2.2.2. Complexité.....	28
I.2.2.3. Les classes d’ordonnement.....	30
I.2.2.4. Les productions statique, dynamique et cyclique	32
• Production statique.....	32
• Production dynamique.....	33
• Production cyclique.....	33
I.3. Modélisation des problèmes d'ordonnement	34
I.3.1. La modélisation mathématique	34
I.3.2. La modélisation graphique	35
I.3.2.1. Graphe Potentiel-Tâches	35

I.3.2.2.	Méthode Pert (Program Evaluation and Research Task).....	36
I.3.2.3.	Réseaux de Petri.....	37
•	Définitions fondamentales.....	37
•	Réseaux de Petri temporisés.....	38
•	Réseaux de Petri temporels	39
I.4.	Méthodes de résolution : état de l’art	41
I.4.1.	Hoist Scheduling Problem.....	41
I.4.1.1.	Problème dynamique	41
•	Pilotage périodique.....	42
•	Pilotage réactif sans prévision.....	42
•	Pilotage réactif avec prévision	43
I.4.1.2.	Problème statique.....	44
•	Branch and Bound.....	46
•	Algorithmes génétiques.....	46
I.4.2.	Généralisation.....	46
I.5.	Position du problème.....	47
I.6.	Conclusion.....	48
II	Chapitre II : Méthodes de résolution du problème SHMP illustré par une ligne de traitement de surfaces	49
II.1.	Introduction	49
II.2.	Description de l’atelier de traitement de surfaces étudié	51
II.2.1.	Contexte de production	51
II.2.2.	Les ressources du système.....	52
II.2.3.	Les opérations de traitement	56
II.2.4.	Temporisation par cycle	57
II.3.	Modélisation par RdPT	59
II.3.1.	Gammes logiques	59
II.3.2.	Gammes opératoires restreintes	61
II.4.	Algorithme de Satisfaction de Contraintes (ASC)	66
II.4.1.	Travaux antérieurs.....	67

II.4.1.1.	Algorithmes de Yih.....	67
II.4.1.2.	Algorithmes de Hindi.....	67
II.4.2.	Formulation du problème	68
II.4.3.	L’algorithme de satisfaction de contraintes proposé.....	72
II.4.3.1.	Description de l’algorithme	73
II.4.3.2.	Résultats obtenus	76
II.4.4.	L’algorithme de satisfaction de contraintes proposé avec hybridation	83
II.4.4.1.	ASC versus ASC / LPTF	85
II.4.4.2.	ASC versus ASC / SPTF.....	88
II.5.	Optimisation utilisant les Algorithmes Génétiques (AGs).....	92
II.5.1.	Terminologie	92
II.5.2.	Fonctionnement des AGs	94
II.5.2.1.	Le codage dans les ateliers de type Jop Shop	94
II.5.2.2.	Opérateur de sélection.....	95
II.5.2.3.	Opérateur de croisement	95
II.5.2.4.	Opérateur de Mutation	98
•	Mutation à un point	98
•	Mutation d’échange ou mutation à 2 points	98
II.5.3.	L’algorithme génétique proposé.....	99
II.5.3.1.	Génération de la population	99
II.5.3.2.	Codage proposé.....	99
II.5.3.3.	Opérateurs proposés.....	100
•	Opérateur de sélection.....	100
•	Opérateur de croisement UPX.....	101
•	Opérateur de mutation à un point	105
II.5.3.4.	Description de l’algorithme	106
II.5.3.5.	Résultats et simulation	110
II.6.	Conclusion.....	114
III	Chapitre III : Approche proposée pour la caractérisation de la robustesse d’un ordonnancement d’ateliers de traitement de surfaces.....	115
III.1.	Introduction	115

III.2.	Robustesse, sensibilité et flexibilité pour les problèmes d'ordonnancement.....	116
III.2.1.	Historique	116
III.2.2.	Terminologie	117
III.2.2.1.	Flexibilité.....	117
III.2.2.2.	Robustesse	118
III.2.2.3.	Flexibilité et robustesse	119
III.2.3.	Classification des perturbations.....	119
III.2.3.1.	Perturbations sur les ressources.....	121
III.2.3.2.	Perturbations sur les travaux.....	121
III.2.4.	Classification des méthodes d'ordonnancement en présence de perturbations ..	122
III.2.4.1.	Les approches réactives	123
III.2.4.2.	Les approches proactives ou robustes	123
III.2.4.3.	Les approches proactives réactives.....	124
III.3.	Modèle pour l'étude de la robustesse pour le problème SHMP.....	124
III.3.1.	Contexte de l'étude.....	125
III.3.2.	Présentation globale de l'approche.....	126
III.3.3.	Eléments de la robustesse	127
III.3.3.1.	Caractéristiques	127
III.3.3.2.	Perturbations.....	128
III.3.3.3.	Mesures de robustesse proposées	128
	• Mesure 1 : Temps de cycle TC	129
	• Mesure 2 : Ecart entre le temps effectif de séjour et la marge minimale	129
III.4.	Exemple de mise en œuvre : Approche par scénarii	130
III.4.1.	1 ^{ère} perturbation : dégradation de la vitesse.....	131
III.4.1.1.	Scénario1 : $TC=6785,2s$	131
III.4.1.2.	Scénario 2 : $TC=6609s$	136
III.4.1.3.	Scénario 3 : $TC=6564,9s$	138
III.4.1.4.	Scénario 4 : $TC=7098,1s$	140
III.4.2.	2 ^{ème} perturbation : diminution des marges maximales	143
III.4.2.1.	Scénario1 : $TC=6785,2s$	143
III.4.2.2.	Scénario 2 : $TC=6609s$	145

III.4.2.3. Scénario 3 : TC=6564,9s	146
III.4.2.4. Scénario 4 : TC=7098,1s	147
III.4.2.5. Etude comparative	147
III.5. Evaluation des performances des solutions.....	148
III.5.1. Formulation des critères	148
III.5.2. Approche d'évaluation multi-critères par l'intégrale de Choquet proposée	150
III.5.3. Algorithme d'évaluation proposé	153
III.5.4. Résultats de mise en oeuvre.....	156
III.6. Conclusion.....	159
CONCLUSION GENERALE	161
BIBLIOGRAPHIE	163

LISTE DES TABLEAUX

Tableau II-1-Déplacements élémentaires du chariot	53
Tableau II-2-Détails des différentes cuves de la ligne de traitement de surfaces	55
Tableau II-3-Cycle de zingage	57
Tableau II-4-Cycle de nickelage	58
Tableau II-5-Cycle d'argentage	58
Tableau II-6-Les temps d'attente du chariot par application de l' ASC	78
Tableau II-7-Séquencement des opérations par application de l' ASC	80
Tableau II-8-Séquencement des cuves par application de l' ASC	80
Tableau II-9-Data-Ordonnancement obtenu par application de l' ASC	81
Tableau II-10-Les temps d'attente du chariot par applicaion de l' ASC/LPTF	86
Tableau II-11-Séquencement des opérations par application de l' ASC/LPTF	87
Tableau II-12-Séquencement des cuves par application de l' ASC/LPTF.....	87
Tableau II-13-Les temps d'attente par application de l' ASC/SPTF	89
Tableau II-14-Séquencement des opérations par application de l' ASC/SPTF	91
Tableau II-15-Séquencement des cuves par application de l' ASC/SPTF.....	91
Tableau II-16-Analogies entre la génétique naturelle et les problèmes d'ordonnancement	93
Tableau II-17-Data-Ordonnancement obtenu par application de l'AG	112
Tableau III-1-Séquencement des opérations relatif au scénario S_{11}	131
Tableau III-2-Séquencement des cuves relatif au scénario S_{11}	132
Tableau III-3-Valeurs des temps d'attente δ_{ji}^k relatifs au scénario S_{11}	132
Tableau III-4-Valeurs des M_{ji}^k relatifs au scénario S_{11}	132
Tableau III-5-Calcul des mesures de robustesse relatif au scénario S_1	134
Tableau III-6-Temps d'attente relatif au scénario $S_{1,17}$	135
Tableau III-7-Temps d'attente relatif au scénario $S_{1,18}$	135
Tableau III-8-Calcul des mesures de robustesse relatif au scénario S_2	137
Tableau III-9-Calcul des mesures de robustesse relatif au scénario S_3	139

Tableau III-10-Calcul des mesures de robustesse relatif au scénario S_4	141
Tableau III-11-Calcul des mesures de robustesse relatif au scénario S_1	144
Tableau III-12-Calcul des mesures de robustesse relatif au scénario S_2	145
Tableau III-13-Calcul des mesures de robustesse relatif au scénario S_3	146
Tableau III-14-Calcul des mesures de robustesse relatif au scénario S_4	147
Tableau III-15-Calcul des indicateurs de performance	149
Tableau III-16-Valeurs des scores de l'intégrale de Choquet	152

LISTE DES FIGURES

Figure I-1-La ligne de traitement de surfaces	23
Figure I-2-Phases du mouvement d'un robot.....	24
Figure I-3-Classes des ordonnancements.....	31
Figure I-4-Exemple de programmation mathématique	35
Figure I-5-Graphe Potentiel-Tâches.....	36
Figure II-1-Gammes logiques associées à l'atelier de traitement de surfaces étudié.....	60
Figure II-2-Gamme opératoire restreinte temporisée, Argentage	57
Figure II-3-Modèle de la charge du travail d'un cycle de production du régime permanent sur un horizon de production discret minimal (=3).....	65
Figure II-4-Temps d'attente du robot par application de l'ASC.....	77
Figure II-5-Variations de la charge des bains, de la charge du robot et du temps du cycle.....	79
Figure II-6-Variations de la charge des bains, de la charge du robot et du temps du cycle pour ASC/LPTF.....	85
Figure II-7-Les temps d'attente du robot par application de l'ASC/SPTF	88
Figure II-8-Variations de la charge des bains, charge du robot et du temps du cycle pour ASC/SPTF.....	90
Figure II-9-Croisement basé sur les positions UPX.....	97
Figure II-10-Présentation des 2 parents	102
Figure II-11-Copie du Parent 1 vers Enfant 1	103
Figure II-12-Suppression des positions sélectionnées dans Parent 2.....	103
Figure II-13-Remplissage des trous de l'enfant 1	104
Figure II-14-Cas d'un ordonnancement non faisable.....	105
Figure II-15-Fonctionnement de l'opérateur de mutation à un point.....	106
Figure II-16-Evolution du temps de cycle à travers les générations pour le problème d'ordonnancement associé à l'atelier de traitement de surfaces étudié.....	111
Figure III-1-Mesures de robustesse relatives au scénario S_1	136
Figure III-2-Mesures de robustesse relatives au scénario S_2	138

Figure III-3-Mesures de robustesse relatives au scénario S_3	140
Figure III-4-Mesures de robustesse relatives au scénario S_4	142
Figure III-5-Variation du temps du cycle pour R implémentations	156
Figure III-6-Le vecteur indices	157

INTRODUCTION GENERALE

Les problèmes d'ordonnancement sont connus pour être complexes et fortement combinatoires et les problèmes de planification de projets sont combinatoires et de complexité polynomiale [Carlier et al, 88].

L'introduction des ressources de transformation relatives aux différentes opérations peut les rendre NP-difficiles dans la majorité des cas.

Deux classes de problèmes d'ordonnancement se présentent dans le contexte de l'ordonnancement cyclique . La première est le problème d'ordonnancement de robot de galvanoplastie, connu sous le nom de Hoist Scheduling Problem ou HSP, la deuxième s'attache à la production flexible manufacturière de type atelier à cheminement multiple également appelé Job Shop.

Le problème HSP peut inclure différents types de problèmes relatifs aux cas [Hindi et al, 04] : mono-robot/mono-produit, mono-robot/multi-produits, multi-robots/mono-produit et multi-robots/multi-produits, alors que pour la galvanoplastie considérée, le problème concerne l'organisation d'une production essentiellement contrainte par une ressource de transport constitué d'un seul robot.

Sur la ligne de traitement de surfaces considérée dans nos travaux, les produits sont immergés successivement dans une série de cuves mono-bac. La durée de traitement de chaque opération a la particularité d'être comprise dans un intervalle, présentant une borne minimale et une borne maximale, appelé "fenêtre de temps". De plus, les produits sont transférés, d'une cuve mono-bac à une autre par un robot à capacité unitaire nécessitant de prendre en compte ces contraintes ainsi que le temps mis pour effectuer ces déplacements pour la caractérisation de l'ordonnancement.

Ce mémoire s'attache à une production multi-produits; en effet, j produits entrent par cycle en ligne à l'état brut et j produits finis sont retirés de la ligne. Dans cette configuration, chaque produit a sa propre gamme constituée d'un ensemble d'opérations. La solution du problème étant cyclique, le calcul de l'ordonnancement optimal relatif au Hoist Scheduling Problem consiste à trouver les mouvements du robot qui minimisent, entre autres la durée du cycle, maximisant

ainsi le taux de production.

La taille et la complexité du problème d'ordonnancement entrent en considération une fois la (ou les) fonction(s) à optimiser est (sont) déterminée(s), pour le choix de la méthode de résolution. Si le problème est de petite taille et de complexité réduite, l'application de l'une des méthodes exactes est suffisante pour aboutir à une solution optimale.

Dans le cas contraire, les méthodes approchées et les techniques d'hybridation constituent le recours le plus efficace pour se rapprocher le plus possible d'une solution optimale.

La résolution d'un problème d'ordonnancement est ainsi constituée par deux phases principales. La première consiste à identifier et à modéliser le problème en décrivant les contraintes et les critères à optimiser alors que la deuxième phase se traduit par la recherche de la méthode adéquate pour résoudre le problème considéré. En effet, l'exploitation de plusieurs méthodes est importante afin de trouver une solution qui permet, par la comparaison de leurs efficacités de générer une solution la plus satisfaisante possible.

C'est dans ce contexte que s'orientent nos travaux de recherche. Ils concernent la résolution d'un atelier réel de type job shop mono-robot/multi-produits dans les lignes de galvanoplastie. Assurer la production en quantité souhaitée et en qualité irréprochable, dans les délais imposés et en respectant les différentes contraintes temporelles et celles de précédence, constitue la complexité de ce type de problème.

L'objectif principal est de construire un ordonnancement cyclique proche de l'optimum d'un point de vue production devant inclure également des critères de robustesse. En effet, le problème est scindé en 2 parties : nous nous sommes focalisés en premier sur le critère performance du temps du cycle. Une évaluation de ces solutions d'un point de vue robustesse est ensuite effectuée et pour terminer nous avons proposé une approche qui intègre dès le départ, c'est-à-dire pendant la phase d'ordonnancement, les critères de robustesse.

Nous présentons alors dans le premier chapitre de ce mémoire, la description et le pilotage des lignes de traitement de surfaces ainsi que les méthodes de modélisation adaptées à ce type de problème. Nous proposons, ensuite, un état de l'art des travaux portant sur les problèmes d'ordonnancement des ateliers en ligne et en particulier le problème du hoist scheduling problem. Trois méthodes de résolution, à savoir l'algorithme de satisfaction de contraintes

proposé qui conduit à un ensemble de solutions, celui-ci hybridé pour améliorer les résultats obtenus et l'algorithme génétique qui assure l'optimisation, ont été appliquées, dans le

deuxième chapitre, pour construire divers ordonnancements de performances diverses. Le troisième chapitre est entièrement consacré à l'étude de la robustesse des ateliers de traitements de surfaces. Nous présentons, d'abord, les différents éléments de la démarche suivie, le type de perturbation considérée, les mesures de robustesse et les indicateurs de performance à travers le modèle que nous proposons pour cette étude. Un algorithme se basant sur une évaluation multicritère en appliquant l'intégrale de Choquet pour le choix systématique d'une solution robuste est enfin développé.

Chapitre I : Problème d'ordonnancement des lignes de traitement de surfaces - Contraintes & Méthodes de résolution

1.1. Introduction

Les problèmes d'ordonnancement sont présents dans tous les secteurs d'activité de l'économie depuis l'informatique, l'agroalimentaire jusqu'à l'industrie manufacturière.

De nombreux ateliers de l'industrie, en particulier ceux de traitement de surfaces, exigent l'achèvement d'une série de tâches tout en respectant :

- les contraintes de précédence qui imposent que certaines tâches doivent être terminées avant que d'autres soient effectuées;
- les contraintes temporelles qui exigent que chaque opération a un temps de traitement compris entre une borne minimale et une borne maximale;
- et les contraintes de ressources qui signifient que deux tâches nécessitent la même ressource et donc ne peuvent se faire simultanément.

La solution est alors de créer un calendrier précisant l'affectation de chaque tâche sur les ressources satisfaisant toutes les contraintes, tout en prenant le moins de temps possible. C'est le problème d'ordonnancement de type job shop qui est NP-complet dans sa forme générale.

Le job shop englobe également le problème du "Hoist Scheduling Problem", relatif à la présence d'un robot chargé de transporter chaque pièce vers les ressources de l'atelier ; ce qui fait croître la complexité de la résolution, si on tient compte des déplacements du robot.

Dans ce chapitre, nous nous intéressons à l'ordonnancement prévisionnel d'un atelier de traitement de surfaces de type job shop traitant plusieurs produits utilisant un seul robot de transport et ayant chacun une séquence d'opérations, donc relatif à un problème "single-hoist/multi-products" (mono-robot/multi-produits).

Après la présentation des ateliers de traitement de surfaces et de leurs spécificités, les outils de

modélisation et les méthodes de résolution, sont, ensuite, introduits. Un état de l'art sur le Hoist Scheduling Problem est enfin présenté.

1.2. Spécificités des lignes de traitement de surfaces

La gestion de la production est directement liée à l'ordonnancement proposé et ceci quelque soit le type d'atelier (atelier de montage, atelier de fabrication mécanique, atelier d'usinage. . .).

Parmi les ateliers existants, les ateliers de traitement de surfaces ont certaines particularités qui leur sont propres. Les travaux menés dans ce mémoire traitent la gestion de ces ateliers.

1.2.1. Description physique de l'atelier étudié

Pour des applications mécaniques ou électriques, il est souvent nécessaire de modifier l'état surfacique des pièces utilisées. Pour cela, des opérations de type traitement de surfaces sont effectuées. Ces opérations consistent à tremper successivement les pièces dans des cuves comportant des produits agissant sur les caractéristiques du matériau [Collart-Dutilleul, 92], [Mangione, 03].

Ces lignes de production se retrouvent notamment dans l'industrie mécanique, pour des opérations de trempe par exemple, dans l'industrie électronique, pour la fabrication des circuits imprimés ou pour les opérations de dépôt d'argent, de cuivre ou d'autres métaux.

Les traitements de surfaces constituent une phase intermédiaire dans le processus de fabrication d'un produit, entre l'usinage et le montage. Ces types de ligne de production sont composés de plusieurs cuves contenant un (des) produit(s) qui permettent de réaliser les traitements souhaités : attaque acide, rinçage, dépôt de cuivre, d'argent ou même d'or, bains électrolytiques, etc.

La gamme opératoire d'un produit est donc constituée d'un enchaînement de traitements à réaliser dans les cuves correspondantes.

Les pièces sont disposées sur des porteurs qui permettent au robot, installé sur un rail, de faire les déplacements de cuve en cuve. Le robot est aussi appelé palan (hoist en anglais) à cause de cette liberté de mouvement dans une seule dimension. La figure I-1, présente un tronçon d'une ligne de traitement de surfaces.

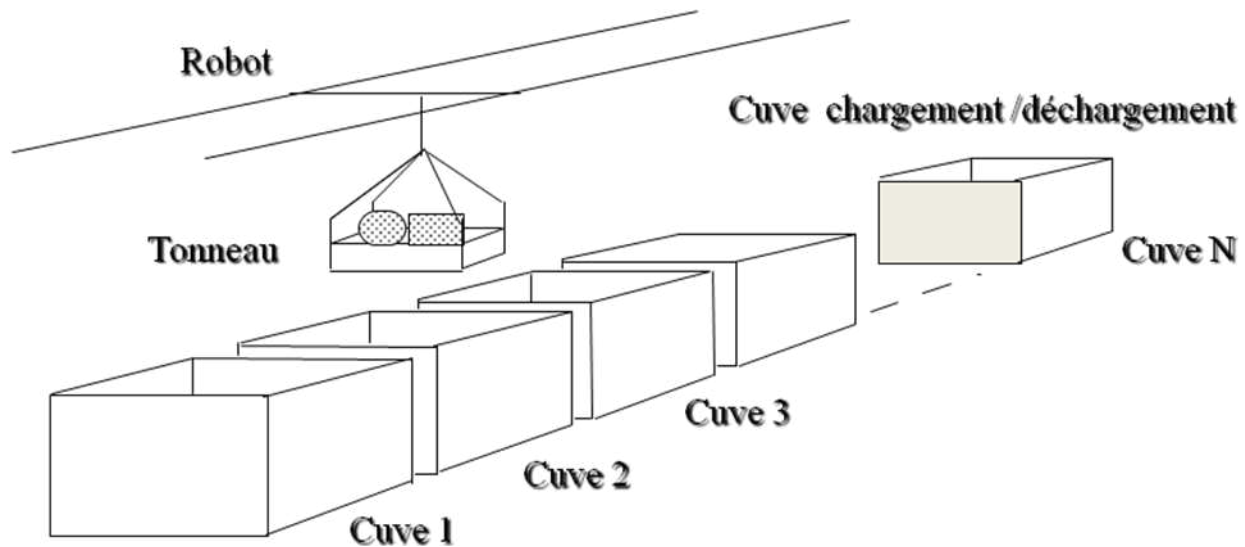


Figure I-1-La ligne de traitement de surfaces

A tout cela, il faut ajouter les systèmes assurant les règles de sécurité et d'hygiène, tels que les systèmes de ventilation et de lavage de fumées, de traitement des eaux, de récupération, etc.

1.2.1.1. Les cuves

Les cuves (ou bassins), servant à recevoir les produits nécessaires aux traitements souhaités sont réalisées dans des matériaux neutres par rapport à leur contenant : polypropylène, fibre de verre, PVC, acier, acier inoxydable, . . .

Chacune des cuves est caractérisée par son utilisation (chargement, déchargement, transfert), par le type de traitement qu'elle contient, par sa capacité et par sa position sur la ligne.

1.2.1.2. Les robots

Les robots ont pour rôle de transporter les porteurs d'une cuve à l'autre. Pour cela, ils se déplacent sur des rails. Chaque robot permet de déplacer les porteurs sur un nombre de cuves soit défini au moment de la conception de la ligne, soit variable en fonction de la production en cours. Les différents phases des mouvements élémentaires effectués par un robot sont, figure I-2, [Mangione, 03] :

-un déplacement à vide, pour se positionner au dessus du porteur à déplacer, phase 1,

-une descente et une saisie pour récupérer le porteur, puis une levée et un égouttage, phase 2,
 -un déplacement en charge,
 -une stabilisation, suivie d'une descente puis d'une phase 3 de libération et enfin une remontée à vide, phase 4. Dans la figure I-2, les flèches en pointillés indiquent les mouvements à vide du robot tandis que celles continues les mouvements en charge.

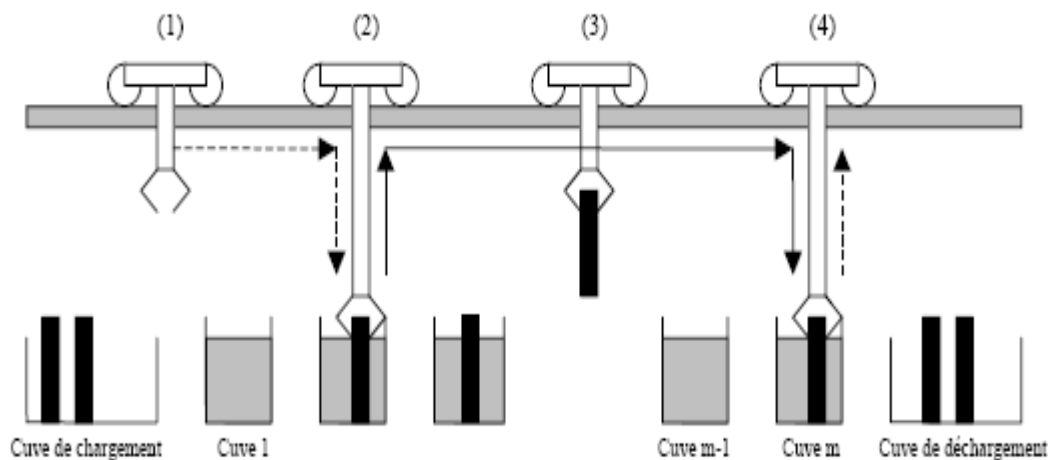


Figure I-2-Phases du mouvement d'un robot

Chaque robot a ses propres caractéristiques dont celles qui servent dans la majeure partie des cas pour le pilotage: la capacité (nombre de porteurs transportables à chaque voyage), les temps de descente, de montée, d'accélération et de décélération, la vitesse maximale et les cuves accessibles.

1.2.2. Le pilotage des lignes de traitement de surfaces

1.2.2.1. Problèmes d'ordonnancement

La gestion des ateliers est un problème récurrent, notamment le fait de savoir qui fait quoi, quand et où. On parle alors de problème d'ordonnancement que l'on peut définir comme suit.

- ***Caractérisation***

Ordonnancement

Ordonnancer c'est affecter des ressources dans le temps pour achever un ensemble d'activités.

Généralement, les problèmes d'ordonnancement d'ateliers de production consistent à affecter, sur les machines (ressources), un ensemble de tâches (opérations ou travaux), elles-mêmes constituées d'un ensemble d'opérations [Rodammer et al, 88]. Chaque opération doit être réalisée sur une machine de l'atelier de production. Mais, l'ordonnancement doit aussi déterminer le séquençement des opérations, en respectant les contraintes liées à l'utilisation des ressources, à la disponibilité des tâches, aux durées d'exécution des opérations. Trouver un bon ordonnancement revient alors à trouver celui qui minimise le critère de performance souhaité (minimisation du coût, de la durée, et du retard ou maximisation de la productivité,...).

Fenêtre de temps pour les durées opératoires

Chaque traitement est constitué de plusieurs immersions dans des cuves contenant des produits différents. La teneur de ces produits est donnée par les ingénieurs chimistes. La différence principale entre une ligne de type traitement de surfaces et une, plus classique, vient du fait que les durées de traitement sont bornées inférieurement et supérieurement. Dans l'exemple où le traitement est une attaque acide, il faut une durée minimale afin que le traitement soit effectué correctement, mais la pièce ne doit pas rester trop longtemps sous peine d'être détériorée et d'entraîner des défauts de qualité.

Les marges sur les temps d'immersion peuvent varier de 0% pour les dépôts de métaux précieux ou les attaques acides jusqu'à des marges infinies, pour des opérations de rinçage où la pièce ne subit aucune dégradation. Cette contrainte sur les durées d'immersion est généralement appelée contrainte de fenêtre de temps (time window constraint).

Le système de transport

Dans certains cas, les systèmes de transport (robot, palans, chariot, . . .) doivent être considérés comme des ressources à part entière. Il faut donc, dans ces cas, non seulement trouver l'ordre des tâches mais aussi ordonnancer les mouvements du (des) robot(s).

- ***Les contraintes***

Les contraintes, liées aux procédés utilisés sur les lignes de traitement de surfaces, nécessitent leur en compte lors de l'élaboration de l'ordonnancement. Le calcul de cet

ordonnancement, tenant compte des contraintes), est alors appelé, dans la littérature, Hoist Scheduling Problem [Shapiro, 85] [Shapiro, 88].

Contrainte de précédence

Suivant le type d'opération à réaliser, il peut être permis, par exemple, de commencer une opération avant que l'opération précédente ne soit terminée. Plusieurs types de précédences peuvent être définis :

- début-début : une opération peut débuter seulement lorsque l'opération précédente a commencé depuis un temps fixé,
- début-fin : une opération peut débuter seulement lorsque l'opération précédente est finie depuis un temps fixé,
- fin-fin : une opération peut finir seulement lorsque l'opération précédente est finie depuis un temps fixé.

Contrainte de sans attente

Une autre contrainte vient s'ajouter à la contrainte de fenêtre de temps, c'est l'impossibilité d'attente entre deux opérations. En effet, pour les lignes de traitement de surfaces, il n'y a pas de possibilité d'avoir des stocks intermédiaires entre les cuves. De plus, un robot, qui vient de récupérer une pièce, ne peut pas attendre un événement (libération d'une cuve ou d'un rail) pour finir les déplacements. Donc, à partir du moment où l'on décide de faire un changement de cuve pour un porteur, on ne peut pas interrompre l'opération (préemption interdite). Pour comprendre cette contrainte, on peut regarder le cas où un porteur sort d'un bain d'acide. Si on le sort du bain et que l'on doit attendre avant de l'immerger dans une cuve de rinçage, l'acide va encore attaquer la pièce et la dégrader entraînant ainsi son rejet en sortie de ligne, pour des raisons de qualité. Cette contrainte est appelée contrainte de sans-attente.

Disponibilité du robot

Une gamme est une succession d'opérations de trempe dans des cuves numérotées d'après l'ordre de passage du premier produit sur la ligne. Une gamme 1-2-3, définie pour un produit, signifie que celui-ci sera trempé respectivement dans les cuves [1], [2] et [3].

Le robot ne peut déplacer un produit de sa cuve que si celui-ci atteint sa durée minimale de traitement autorisée. Le transport d'un produit d'une cuve à une autre suit une gamme de traitement prédéfinie.

Comme les porteurs ont une durée d'immersion bornée dans chacune des cuves, il faut donc qu'à la fin de chaque traitement un robot soit disponible afin de sortir le porteur de sa cuve pour éviter les détériorations ou les coûts supplémentaires éventuels.

Ce problème de disponibilité du robot ainsi que celui des cuves nécessitent de gérer convenablement les déplacements du robot. Le HSP consiste à trouver l'ordre optimal d'entrée des porteurs sur la ligne, mais aussi les durées de trempe effectives pour chacune des opérations et les déplacements des robots.

Disponibilité des cuves

Comme il a été dit précédemment, les cuves peuvent recevoir un porteur (pour des cuves mono-bac), ou plusieurs porteurs (pour des cuves multi-bacs). Il faut donc vérifier avant chaque déplacement de porteur que l'on va pouvoir déposer celui-ci dans la cuve. Il faut donc que la cuve soit déjà vide pour des cuves mono-bac ou qu'il reste un emplacement libre dans les cuves multi-bacs.

- ***Les critères d'optimisation***

Un problème d'ordonnancement n'est pas nécessairement exprimé comme un problème d'optimisation. Néanmoins, la notion de critère d'optimisation est toujours présente, au moins implicitement. Un chef d'atelier veut pouvoir terminer le plan de production journalier dans les délais imposés. Un chef de projet désire assurer une charge de travail la plus constante possible à son équipe durant le déroulement du projet. Un programmeur, utilisant de lourdes ressources de calcul, désire que son application soit correctement exécutée, et que le résultat lui parvienne au bout d'un temps raisonnable. Ainsi, les critères possibles, dépendant de l'application considérée, sont très nombreux. Il peut même y en avoir plusieurs à la fois.

Parmi les ordonnancements possibles, nous devons choisir la solution la plus satisfaisante. Cette notion de satisfaction dépend du critère préalablement défini. Les critères sont donc les barèmes qui vont servir à l'évaluation et à la comparaison de l'ensemble des ordonnancements possibles.

Ces critères peuvent être classés en deux types [Kacem, 03] : réguliers et irréguliers.

Les critères réguliers

Les critères, dits réguliers, constituent des fonctions décroissantes des dates d'achèvement des opérations. Nous en citons à titre d'exemples :

- la minimisation des dates d'achèvement des actions ;
- la minimisation du maximum des dates d'achèvement des actions ;
- la minimisation de la moyenne des dates d'achèvement des actions ;
- la minimisation des retards sur les dates d'achèvement des actions ;
- la minimisation du maximum des retards sur les dates d'achèvement des actions ;
- la minimisation de la moyenne des retards sur les dates d'achèvement des actions ;
- la minimisation du temps du cycle.

Les critères irréguliers

Ces critères, non réguliers, ne sont pas des fonctions monotones des dates de fin d'exécution des opérations. Soit à titre d'exemples :

- la minimisation des encours ;
- la minimisation du coût du stockage des matières premières ;
- l'équilibrage des charges des machines.

1.2.2.2. Complexité

La complexité des problèmes d'ordonnancement est définie suivant la complexité des méthodes de résolution et celle des algorithmes utilisés [Tangour, 07]. Certains problèmes d'ordonnancement de taille relativement importante peuvent avoir un niveau de combinatoire si important que leur résolution devient très difficile.

Deux catégories de complexité sont distinguées : algorithmique et problématique.

- **Complexité problématique**

La complexité problématique dépend du problème à résoudre ainsi que de la méthode de résolution choisie pour construire la solution optimale au sens des critères retenus. Dans la

littérature, les problèmes d'ordonnement sont souvent classés en deux catégories caractérisant leurs degrés de complexité :

-Les problèmes indécidables qui sont les problèmes d'ordonnement les plus difficiles pour lesquels il n'existe aucune méthode de résolution,

-les problèmes décidables qui sont de classe P ou de classe NP [Charon et al, 96].

-Un problème de décision est un problème qui comprend deux parties : une partie donnée du problème et une question binaire ayant « oui » ou « non » comme réponse possible.

-Un problème de recherche est un problème constitué d'un ensemble de données et pour lequel à chaque ensemble de données correspond un ensemble de solutions. Résoudre un problème de recherche consiste à calculer, pour chaque ensemble de données D, l'ensemble des solutions S (D) associées.

-Un problème d'optimisation est un problème de recherche qui donne à chaque solution une valeur quantitative. On cherche une valeur minimale si on a une fonction économique à minimiser, ou maximale pour d'autres cas.

À chaque problème d'optimisation on peut associer un problème de décision [Charon et al, 96]. Ainsi, l'étude de la complexité d'un problème de décision permet de donner les indications relatives au problème d'optimisation associé.

-Un problème de décision est de classe P s'il existe un algorithme polynomial pour le résoudre.

-Un problème de décision est de classe NP, dit également NP-difficile, s'il ne peut pas être résolu en un temps polynomial par les algorithmes déterministes [Carlier et al, 88], mais peut être résolu en un temps polynomial par des méthodes approchées.

-Un problème de décision est dit NP-complet s'il appartient à la classe NP et s'il est résolu, au mieux, en un temps exponentiel.

-Un problème d'optimisation est dit NP-difficile si le problème de décision associé est NP-complet.

- **Complexité algorithmique**

La théorie de la complexité a pour objectif d'analyser les coûts de résolution, notamment en termes de temps de calcul, des problèmes d'optimisation combinatoire. Elle permet d'établir une classification des problèmes en plusieurs niveaux de difficulté, et fait la distinction entre un

problème d'optimisation et un problème de décision [Carlier et al, 88]. IL a été démontré que la plupart des problèmes d'ordonnancement sont difficiles [Lopez et al, 00].

La complexité algorithmique se mesure par rapport au temps alloué pour l'exécution de l'algorithme ou encore par rapport à l'espace mémoire requis. Le temps de calcul est fonction du nombre d'instructions et de la taille des données manipulées.

1.2.2.3. Les classes d'ordonnancement

En se référant à la littérature [Carlier et al, 88], [Saad, 07], [Tangour, 07], une classification des problèmes d'ordonnancement peut s'établir selon plusieurs critères :

- le changement d'une opération ;
- le nombre de machines et leur ordre d'utilisation pour fabriquer un produit ;
- et en fonction de l'enchaînement des opérations.

Il existe différentes classes d'ordonnancement selon le premier critère [Saad, 07].

-Dans un ordonnancement semi-actif : il est impossible d'avancer une opération sans modifier la séquence des opérations sur la ressource : chaque opération est calée, soit sur l'opération qui la précède dans sa gamme, soit sur l'opération qui la précède sur la machine utilisée.

-Dans un ordonnancement actif : il est impossible d'avancer une opération sans reporter le début d'une autre opération.

-Dans ordonnancement sans délai ou sans retard : aucune opération n'est mise en attente, alors qu'une machine est disponible pour l'exécuter.

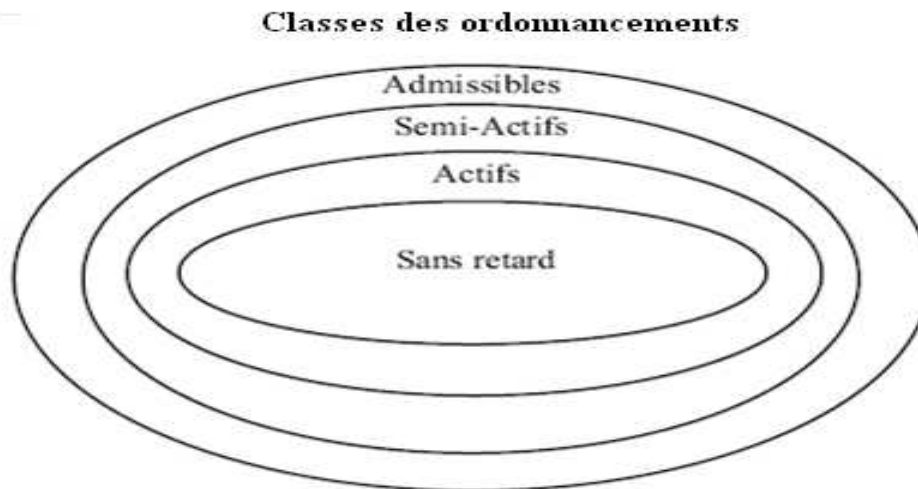


Figure I-3-Classes des ordonnancements

La figure I-3 présente le diagramme d'inclusion des classes d'ordonnancements. Elle met en exergue que 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.

Une deuxième classification peut se faire selon le deuxième critère; en effet, un atelier est caractérisé par le nombre de machines qu'il contient et par son type. Pour les différents types d'ateliers possibles, différents problèmes peuvent être posés [Carlier et al, 88].

-Problème à une machine : chaque produit est constitué d'une seule opération. L'intérêt de ce type de problème réside dans le fait qu'il permet de développer des méthodes utilisables pour la résolution de problèmes plus complexes.

-Problème à machines parallèles : elles remplissent, a priori, toutes les mêmes fonctions.

- des machines identiques : la vitesse d'exécution est la même pour toutes les machines et pour tous les produits ;
- des machines uniformes : chaque machine a une vitesse d'exécution propre et constante pour tous les produits (d'une même machine) ;

-
- des machines indépendantes : la vitesse d'exécution est différente pour chaque machine et pour chaque produit.

Une dernière classification peut se faire en fonction de l'enchaînement des opérations.

-Problème de flow shop : dans les ateliers de type flow shop, la ligne de fabrication est constituée de plusieurs machines en série ; toutes les opérations de tous les produits passent par toutes les machines dans le même et unique ordre. Ce type d'atelier est dit à cheminement unique.

La littérature présente une extension possible de ce problème : le flow shop hybride pour lequel les machines sont disponibles en plusieurs exemplaires.

-Problème de job shop : il constitue une généralisation directe de celui du flow shop. Dans les ateliers de type job shop, les opérations sont réalisées selon un ordre total bien déterminé, variant selon le produit à exécuter. Ce type d'atelier est nommé aussi atelier à cheminements multiples. Dans ce cas, plusieurs changements d'outils sont à envisager.

Le job-shop flexible est une extension du modèle job shop classique. Sa particularité essentielle réside dans le fait que plusieurs machines sont potentiellement capables de réaliser un sous-ensemble d'opérations. Plus précisément, une opération est associée à un ensemble contenant toutes les machines pouvant effectuer cette opération. Ce type de problème est celui traité dans ce mémoire.

-Problème de l'open shop : Aucun ordre de fabrication n'est imposé, dans ce cas ; l'acheminement de toutes les opérations est multiple et libre; ces opérations peuvent être exécutées dans n'importe quel ordre.

Suivant le nombre et le type des produits à traiter, on distingue différents types de production, par conséquent les méthodes de gestion de la ligne diffèrent, [Mangione, 03].

1.2.2.4. Les productions statique, dynamique et cyclique

- ***Production statique***

Les pièces à produire sont connues à l'avance pour une période donnée. Il faut alors trouver l'ordre et le moment pour faire entrer les pièces dans la ligne ainsi que les durées effectives de trempe et les mouvements du robot.

- *Production dynamique*

Les pièces à produire ne sont pas connues à l'avance mais arrivent "au fil de l'eau". Il faut alors décider, lorsque de nouvelles pièces arrivent, si l'on doit commencer à traiter la pièce ou s'il faut attendre ; tout ceci dépend de l'état du système et des arrivées de pièces éventuelles.

- *Production cyclique*

C'est un cas particulier de la production statique pour laquelle un critère est recherché. En effet, les pièces entrent dans le système l'une après l'autre. Le robot exécute une séquence fixe des mouvements de manière répétitive. Chaque répétition est appelée cycle ou période. La durée d'un cycle est appelée "temps de cycle". La séquence des mouvements donne un ordonnancement cyclique. La minimisation du temps de cycle en cherchant un ordonnancement cyclique optimal est un des points clés pour maximiser la productivité de la ligne [Che, 2001].

Nous étudions en particulier le problème multi-cycliques ou r-cycliques dans le cas où r pièces sont introduites dans le système (pour la conservation de flux, r pièces sont nécessairement retirées de la ligne) à chaque période.

De nombreux chercheurs se sont intéressés à l'ordonnancement cyclique simple ($r=1$) [Chen et al, 94], [Kats et al, 97], [Song et al, 97], [Levner et al, 98]. L'ordonnancement multi-cycliques est, en revanche, relativement peu étudié. Un algorithme par séparation et évaluation a été développé pour calculer un ordonnancement 2-cycliques optimal. [Lei et al, 94]. Dans [Song et al, 97] est proposé un modèle de programmation linéaire mixte pour l'ordonnancement dans un processus chimique sans attente. Un algorithme, basé sur des méthodes filtrantes pour l'ordonnancement multi-cycliques dans un système sans attente, a été proposé par [Kats et al, 99]. Considérant que cette méthode se limite à la recherche des dates d'entrée entières des pièces et d'un temps de cycle entier ; elle ne garantit pas l'optimalité des solutions trouvées.

La cyclicité peut être définie autrement, [Mangione, 03] ; la ligne produit généralement les mêmes types de pièces et dans des quantités qui varient peu sur la période étudiée. Ce type de production est, par exemple, utilisé lorsque l'entreprise décide de produire par campagne.

Cette approche peut s'appliquer, aussi, quand les gammes de traitement entre les différents types

de produits sont les mêmes et que les temps de trempe sont très proches. Il est alors possible de considérer tous les types de produits comme un seul lors de l'ordonnement.

Le but de l'ordonnement est de définir le cycle de production optimisant un (ou plusieurs) critère(s) qui peut (vent) être la maximisation de la productivité ou (et) la minimisation de l'encours par exemple.

1.3. Modélisation des problèmes d'ordonnement

La modélisation est généralement une étape très importante de la résolution d'un problème. C'est une écriture simplifiée de toutes les données tout en utilisant un formalisme bien adapté pour représenter un problème choisi. Dans la littérature, on trouve principalement deux méthodes pour modéliser les problèmes d'ordonnement : les méthodes mathématiques et les méthodes graphiques.

1.3.1. La modélisation mathématique

Ces méthodes consistent à représenter les données du problème sous forme d'équations et d'inéquations mathématiques, [Saad, 07].

Très couramment utilisées, elles ont l'avantage d'être simples d'une part et directement exploitables par les algorithmes de résolution d'autre part [Lopez et al, 99].

Exemple I-1

Soit un ensemble de 7 tâches, figure I-4, il est question de calculer t_i pour $i = 1, \dots, 7$, la date de début d'exécution de la tâche i , en minimisant le C_{\max} et en respectant les contraintes, d_i étant la durée opératoire de la tâche i .

<i>Contraintes des données</i>	<i>Contraintes de précédence</i>	<i>Contraintes des ressources</i>
$d_1 = 3$	$t_1 + d_1 \leq t_3$	$(t_1 + d_1 \leq t_2) OU (t_2 + d_2 \leq t_1)$
$d_2 = 2$	$t_1 + d_1 \leq t_4$	
$d_3 = 4$		$(t_3 + d_3 \leq t_5) OU (t_5 + d_5 \leq t_3)$
$d_4 = 1$	$t_3 + d_3 \leq t_6$	$(t_3 + d_3 \leq t_7) OU (t_7 + d_7 \leq t_3)$
$d_5 = 8$	$t_4 + d_4 \leq t_7$	$(t_7 + d_7 \leq t_5) OU (t_5 + d_5 \leq t_7)$
$d_6 = 3$	$t_2 + d_2 \leq t_5$	
$d_7 = 7$		$(t_4 + d_4 \leq t_6) OU (t_6 + d_6 \leq t_4)$

Figure I-4-Exemple de programmation mathématique

1.3.2. La modélisation graphique

1.3.2.1. Graphe Potentiel-Tâches

Pour modéliser un problème d'ordonnement, nous pouvons avoir recours aux graphes. Dans ce genre de modélisation, les tâches sont représentées par des nœuds, les contraintes de précédence par des arcs conjonctifs tout en indiquant les durées des tâches et finalement les contraintes de ressources par des arcs disjonctifs [Roy, 70], [Gotha, 93].

Un problème d'ordonnement peut être représenté par un graphe potentiel-tâches noté $G(X;U)$, où X est l'ensemble des sommets et U l'ensemble des arcs.

Si on appelle T l'ensemble des n tâches du problème, X est alors la réunion de T et de l'ensemble $(0, n+1)$ où (0) représente la tâche du début fictif et $\{n+1\}$ la tâche de la fin fictive.

L'ensemble des arcs U est $\{(0; i), (i; j) \text{ et } (i; n+1) \text{ tels que } i \text{ et } j \in T\}$. Il représente les différentes contraintes reliant les tâches, Fig. I-5.

Les valeurs portées par les arcs dépendent des sommets :

- l'arc de type $(0; i)$: porte la date de disponibilité de la tâche i ;

- l'arc de types $(i; j)$: porte la valeur de la durée opératoire de la tâche i , un deuxième arc portant une valeur négative peut éventuellement relier j et i pour visualiser le fait que la tâche j doit commencer immédiatement après la tâche i ;
- l'arc de type $(i; n)$: porte la durée opératoire de i .

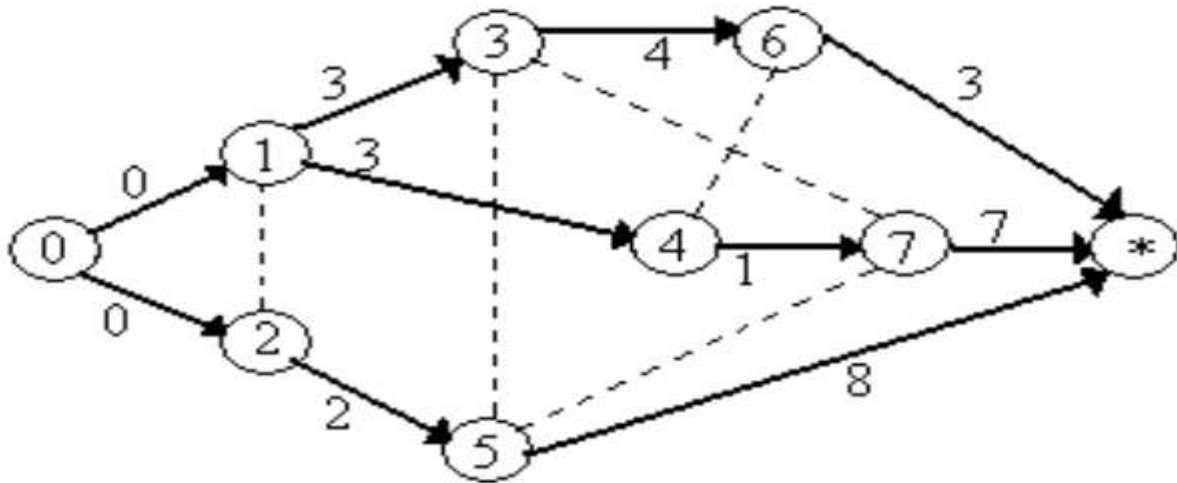


Figure I-5-Graphe Potentiel-Tâches

La figure I-5, [kacem, 03], constitue la modélisation d'un problème d'ordonnancement de 7 tâches à exécuter sur 3 ressources. Les contraintes de précédence sont les suivantes : 1 avant 3 ; 3 avant 6 ; 1 avant 4 ; 4 avant 7 ; 2 avant 5. Les durées des tâches sont indiquées sur les arcs conjonctifs (par exemple la durée de la tâche 1 est de 3 unités de temps). Les tâches 1 et 2 doivent être réalisées sur la première ressource. Les tâches 3, 5 et 7 doivent être réalisées sur la deuxième ressource. La troisième ressource exécutera les tâches 4 et 6. Ainsi, les contraintes sur les ressources sont représentées par les arêtes pointillées. Le fait de choisir un sens pour ces arêtes donne un ordonnancement possible.

I.3.2.2. Méthode Pert (Program Evaluation and Research Task)

Cette représentation, semblable à la précédente, permet de représenter une tâche par un arc, auquel est associé un chiffre qui représente la durée de la tâche. Entre les arcs figurent des cercles, appelés sommets ou événements, qui marquent l'aboutissement d'une ou de plusieurs tâches. Ces cercles sont numérotés afin de suivre l'ordre de succession des divers événements.

1.3.2.3. Réseaux de Petri

Introduits par Carl Adam Petri en 1962, les RdPs constituent un puissant outil de modélisation graphique de représentation de phénomènes complexes et des mécanismes séquentiels. Ils sont largement utilisés pour l'analyse et la modélisation des systèmes à événements discrets [Murata, 89].

- *Définitions fondamentales*

Définition 1

Un RdP non marqué est un graphe orienté biparti constitué de places, de transitions (correspondant aux sommets du graphe) et d'arcs qui relient les transitions aux places et les places aux transitions. Il est représenté par un quadruplet $Q = \langle P, T, I, O \rangle$ tel que :

P : est un ensemble fini et non vide de places;

T : est un ensemble fini et non vide de transitions;

$P \cap T = \emptyset$: les ensembles P et T sont disjoints;

I : est l'application d'incidence avant : $I : P \times T \rightarrow \mathbb{N}$ (ensemble des entiers naturels), correspondant aux arcs directs des places vers les transitions ;

$I(p,t) > 0$ signifie qu'il existe un arc orienté de p vers t et $I(p,t)$ est la validation de cet arc ;

$I(p,t) = 0$ indique l'absence d'arc orienté reliant la place p à la transition t ;

O : est l'application d'incidence arrière : $O : T \times P \rightarrow \mathbb{N}$ correspondant aux arcs directs liant les transitions aux places ;

$O(t,p) > 0$ signifie qu'il existe un arc orienté de t vers p et $O(t,p)$ est la validation de cet arc ;

$O(t,p) = 0$ indique l'absence d'arc orienté reliant la transition t à la place p .

Si $O(t,p) > 0$ (respectivement $I(p,t) > 0$), la transition t est une transition de sortie (respectivement d'entrée) de la place p . La place p est dite place d'entrée (respectivement de sortie) de la transition t .

Lorsque les applications I et O prennent leurs valeurs dans la paire $\{0, 1\}$, le réseau est dit

ordinaire. Dans la suite, nous supposons que le RdP considéré est ordinaire.

Définition 2

Un RdP marqué est un doublet $R = \langle R', M_0 \rangle$ dans lequel R' est un RdP non marqué et M_0 un marquage initial.

Dans un RdP marqué, chaque place contient un nombre entier (positif ou nul) de marques ou jetons. Le nombre de marques contenu dans une place p_i est noté $M(p_i)$. Le marquage du réseau M est défini par le vecteur de ces marquages. Le marquage à un certain instant définit l'état du RdP, ou plus précisément l'état du système décrit par le RdP.

Définition 3

Une place p est dite bornée ou k -bornée pour un marquage initial M_0 s'il existe un entier naturel k , tel que pour tout marquage accessible M , le nombre de marques dans p reste inférieur ou égal à k .

Définition 4

Un RdP est k -borné pour un marquage initial M_0 , si toutes les places sont k -bornées pour M_0 . Un RdP est dit sauf (ou binaire) s'il est 1-borné.

- ***Réseaux de Petri temporisés***

Historiquement, les premiers modèles, qui intégraient le temps, relevaient d'une logique spécifique qui consistait à décrire le temps nécessaire à une opération. Cette approche permet de représenter les mécanismes temporels associés à un processus [Collart-Dutilleul, 08].

Nous sommes donc en face d'un outil performant pour la spécification et la validation des systèmes comportant des temps minimums.

Le temps peut être associé indifféremment aux transitions (modèle RdP t -temporisé) ou aux places (modèle RdP p -temporisé). Nous considérons dans nos développements les réseaux de Petri p -temporisés où le temps est associé aux places.

Définition 5

Un réseau de Petri temporisé avec n places et p transitions est un doublet $RT = \langle R, D \rangle$ avec :

R est un réseau de Petri $\langle P, T, Pre, Post \rangle$ avec un marquage initial M_0 ,

D est la fonction durée minimale de séjour d'une marque dans une place donnée :

$D : P \longrightarrow Q+$

qui à chaque place fait correspondre un nombre rationnel positif décrivant la durée d'indisponibilité des jetons.

La sémantique est que les marques doivent rester dans la place p_i , au moins le temps d_i associé à cette place. Pendant d_i , la marque est indisponible ; elle ne participe pas à la validation des transitions. d_i représente donc :

- la durée d'indisponibilité de la marque pour la validation des transitions,
- le temps minimum de séjour d'une marque dans une place.

- ***Réseaux de Petri temporels***

La production manufacturière et particulièrement les industries faisant appel à des procédés de nature chimique utilisent souvent des opérations dont la durée doit être comprise entre un maximum et un minimum. Dans le cas d'une cuisson dans un four ou d'une trempe dans un bain d'acide, il est évident que le non-respect de ces contraintes entraînera une altération de la qualité du produit [Collart-Dutilleul, 08].

Les RdP temporisés ne modélisent que les contraintes de temps minimum, c'est pourquoi nous utiliserons les RdP P-temporels pour traduire l'obligation de respect des temps de séjour.

Notons bien que les possibilités de spécifications des RdP P-temporels incluent celles des RdPT.

Les modèles temporels de Roux et Ménasche ne traduisent pas cette obligation de temps de séjour sur les synchronisations. Dans ces modèles, l'intervalle de temps est associé aux transitions et on ne commence à compter le temps que lorsque celles-ci sont validées. Ainsi une marque peut rester jusqu'à l'infini dans une place en amont d'une transition de synchronisation (jusqu'à la validation de cette transition). On perd donc l'histoire d'une marque en amont d'une telle transition. En conséquence, les RdP t-temporels ne traduisent pas l'obligation sur les synchronisations (au contraire des autres transitions), montrant ainsi que les RdP t-temporels n'ont pas la puissance de spécification nécessaire à la modélisation de ce type de système. C'est pour cette raison que le modèle RdP P-temporel a été introduit.

Définition 6

Un RdP temporel marqué peut se définir comme un sextuplet $R = (P, T, I, O, It, M)$, tel que :

- P est l'ensemble des places,

- T est l'ensemble des transitions,
- I est la fonction qui définit les arcs en spécifiant les places d'entrée des transitions
 $I : P \times T \rightarrow \{0, 1\}$
- O est la fonction qui définit les arcs en spécifiant les places de sortie des transitions
 $O : P \times T \rightarrow \{0, 1\}$
- M est une application qui associe à chaque place p du RdP un nombre de marques M(p) Entier et, Mo le marquage initial.
- $I_t : P \rightarrow (Q^+ \cup 0) \times (Q^+ \cup \infty)$, (Q^+ étant l'ensemble des nombres rationnels positifs),
 $p_i \rightarrow I_{ti} = [a_i, b_i]$ avec : $0 \leq a_i \leq b_i$
 $\forall i, 1 \leq i \leq n, n = \text{Card}(P)$.

I_{ti} définit l'intervalle de temps de séjour d'une marque dans la place p_i . Remarquons qu'une marque dans la place p_i ne participe à la validation de ses transitions de sortie que quand elle a passé au moins la durée a_i dans cette place et au plus la durée b_i ; après b_i , la marque sera dite "morte".

L'évolution dynamique d'un RdP P-temporel dépend des marques et de leurs situations temporelles : à un instant donné, l'état est totalement déterminé par la paire $E = (M, Q)$, M étant une application de marquage assignant à chaque place du réseau un certain nombre de marques ($\forall p \in P, M(p) \geq 0$) et Q une application de temps de séjour qui associe à chaque couple (place, marque), un nombre rationnel q, q devenant être inférieur ou égal à b_i .

L'utilisation d'un outil de modélisation graphique, tel que les RdPs pour résoudre des problèmes d'ordonnancement, reste encore assez rare dans la littérature. Mais, l'intérêt de pouvoir représenter les gammes de fabrication avec les contraintes de précédence entre les opérations et les contraintes de ressources partagées, a déjà suscité quelques approches [Camus, 97].

Le développement du modèle Réseau de Petri sur lequel va reposer notre ordonnancement cyclique est basé sur la connaissance des gammes de fabrication avec notamment le raffinement progressif de ces gammes : passage d'une gamme logique en une gamme opératoire avec la prise en compte des informations relatives au procédé afin de spécifier sur quelle ressource une

opération est affectée. Par la connaissance des ressources de transformation et des liens physiques associés, nous pouvons commencer à envisager les opérations de transfert entre ces ressources. Cette dualité entre les ressources et les produits (gammes) est parfaitement mise en évidence avec le modèle RdP présenté ultérieurement.

1.4. Méthodes de résolution : état de l'art

Les problèmes d'ordonnancement de la production couvrent un éventail de recherches important. Beaucoup d'ouvrages sont consacrés à ce domaine [Blaewicz et al, 94], [Pinedo, 95], [Lopez et al, 00], [Pujo et al, 02].

En particulier pour les problèmes HSP, il y a globalement, deux cas qui ont été distingués dans la littérature [Subai et al, 03] : cyclique et non cyclique. Une bonne synthèse de travaux correspondants a été présentée dans [Bloch. C. et al, 1997].

En pilotage cyclique, les travaux fondateurs furent ceux de Phillips [Phillips et al, 76]. Un bon état de l'art est présenté dans [Manier et al, 94].

1.4.1. Hoist Scheduling Problem

Dans cette partie, nous nous référons principalement à l'état de l'art réalisé par [Manier et al, 94] puis [Baptiste et al, 01] sur le HSP qui décrit les différentes méthodes de résolution avec leurs avantages et inconvénients.

Ces derniers dépendent essentiellement de la nature du problème à traiter.

1.4.1.1. Problème dynamique

Lorsque la demande est incertaine, il est possible de modéliser les événements aléatoires par des lois stochastiques dans [Fleury et al, 99] et [Fleury et al, 01], l'intérêt a été porté au HSP stochastique en particulier au SHPS, en proposant une méta-heuristique pour trouver une solution avec comme critère la minimisation du Makespan, mise en place lorsque les conséquences des événements aléatoires sont faibles.

En ce qui concerne le problème dynamique, différents modes de pilotage ont été proposés pour le résoudre :

-
- périodique et prédictif ;
 - réactif sans prévision ;
 - réactif avec prévision.

- ***Pilotage périodique***

Le pilotage périodique est utilisé pour des travaux avec un grand horizon de temps (grande série, ou production par campagne). Il consiste à déterminer le cycle optimum pour une suite de traitements, en minimisant le rapport (durée du cycle / nombre de porteurs introduits dans le cycle) avec les outils de type “programmation par contraintes”. L’inconvénient d’un tel mode de pilotage vient des difficultés d’évolution lors d’un changement de gammes ou de pièces. Les méthodes utilisées, dans ces cas, ont pour objectif soit de vider l’atelier avant de lancer la nouvelle gamme, soit d’insérer le nouveau cycle en trois temps [Vanier, 96] :

- la dégradation de l’ancien cycle,
- l’insertion du nouveau cycle dégradé dans l’ancien dégradé,
- dès que l’ancien cycle est terminé, revenir au nouveau cycle optimal.

Dans le cas où il faut insérer une nouvelle opération dans un cycle sans en changer la séquence, la technique développée utilise la théorie des graphes [Artigues et al, 97].

- ***Pilotage réactif sans prévision***

Le pilotage réactif sans prévision est le suivant : après chaque opération de transport, sont recalculés les nouveaux mouvements du robot. Les temps de calcul devant être faibles pour ne pas retarder le robot, on emploie des heuristiques pour trouver les nouveaux mouvements. Le choix des heuristiques à utiliser est donné par un système expert, basé sur la connaissance d’un expert humain; c’est à dire que, suivant la situation de l’atelier, on choisit l’heuristique qui donne la meilleure solution, déterminé au préalable par simulation [Yih, 88]. L’inconvénient d’une telle méthode vient surtout du fait qu’il est très dur de respecter les bornes.

Dans [Chauvet et al, 00], est proposé un algorithme dynamique, qui garantit la date de sortie minimale pour chaque nouveau porteur, et est montré comment généraliser cet algorithme au cas multi-robots.

- ***Pilotage réactif avec prévision***

Le pilotage réactif avec prévision calcule, à chaque arrivée de porteur, un nouvel ordonnancement en tenant compte de ce qui a été déjà fait et ce qui doit être fait (position des porteurs sur la ligne, gamme des porteurs. . .). Ce pilotage a pour avantage de respecter les bornes mais aussi d'être peu sensible aux perturbations.

Son inconvénient, par contre, est qu'il ne permet pas une optimisation sur le long terme. Pour calculer le nouvel ordonnancement quatre heuristiques sont possibles.

–L'heuristique simple [Yih et al, 92] : elle considère la possibilité de faire rentrer un nouveau porteur sur la ligne sans risque de conflits (de cuves ou de robots). Dans le cas d'une réponse positive, elle lance le nouveau porteur, sinon elle décale son entrée sur la ligne. Cette solution a pour particularité de ne pas tenir compte des tolérances sur les durées de séjour dans les cuves. Elle est efficace dans les problèmes où la variance des durées de trempe est faible, le robot rapide et le rapport (moyenne des durées de trempe / vitesse du robot) important.

–L'heuristique avec conservation de l'ordonnancement [Yih et al, 92] : elle consiste à regarder si la tolérance sur le nouveau porteur au niveau du conflit ne permet pas de le supprimer ou de réduire le temps d'attente en entrée.

Le fait de jouer uniquement sur les tolérances du nouveau porteur limite l'utilisation de cette méthode aux cas où le robot est rapide et la tolérance sur le porteur en entrée large.

–L'heuristique avec conservation de la séquence [Yih, 94] : les tolérances des deux porteurs en conflit sont utilisées pour voir si, en modifiant les durées de trempe, on peut régler ces conflits. Par contre, cette méthode ne tient pas compte du fait qu'un décalage d'un porteur sur la ligne puisse entraîner un nouveau conflit.

–L'heuristique sans conservation [Yih et al, 95] : c'est une méthode de type "procédure par séparation et évaluation" où toutes les tolérances sont utilisées pour essayer de résoudre le problème.

Procédure par séparation et évaluation

Une autre méthode pour régler les conflits (ou problème local) a été développée par Lamothe [Lamothe, 96]. Elle consiste à modéliser le problème par contraintes puis le résoudre de manière

arborescente avec une Procédure par Séparation et Evaluation Progressive (PSEP) en profondeur. Pour limiter les temps de calcul, l'auteur met en place des informations d'inconsistance (Nogood) et de mémorisation des raisonnements déjà réalisés (dynamic backtracking) ; ce qui permet de ne pas tout recalculer à chaque problème local.

Algorithme évolutionniste

Un algorithme évolutionniste a été mis au point pour résoudre le cas du problème dynamique [Bloch, 99]. Cet algorithme s'appuyant sur la méthode génétique décrite précédemment, consiste à calculer, à chaque arrivée de nouveaux porteurs, un nouvel ordonnancement en tenant compte des travaux déjà effectués (seuil S1) et des temps de calcul (seuil S2). Pour vérifier la validité des solutions obtenues, un calcul du plus long chemin dans un graphe est utilisé. Une fois une solution trouvée, elle sert d'ordonnancement partiel pour l'itération suivante ; et ainsi de suite jusqu'à l'ordonnancement du dernier travail

1.4.1.2. Problème statique

Le problème du HSP est un problème complexe à résoudre. Il a été montré que trouver l'ordonnancement cyclique optimal d'une ligne comprenant un seul robot, pour une production mono-produit, fait partie de la classe des problèmes NP-complets, pour des temps de transport quelconques [Lei et al, 89] et pour des temps de transport additifs et symétriques [Brauner et al, 97].

Il existe, pour résoudre ce problème statique, les méthodes exactes parmi lesquels on cite la programmation linéaire [Song et al, 97], [Mellouli et al, 04], les techniques de satisfaction de contraintes [Yih, 94], [Baptiste et al, 96], [Cheng et al, 94], [Cheng et al, 97], [Hindi et al, 04], [Mhedhbi et al, 10], et la méthode de type Branch & Bound [Lei et al, 91], [Chen et al, 94], [Chen et al, 95], [Chen et al, 97],

On trouve également les métaheuristiques dont on distingue principalement les algorithmes génétiques [Lim, 97], [Pezzalla et al, 08].

- ***Programmation linéaire***

Le problème est modélisé par un programme linéaire mixte [Song et al, 97]. Comme le modèle obtenu nécessite un nombre trop important de variables et de contraintes, la formulation permet alors de développer une heuristique (EST : Earliest Start Time) pour générer une solution approchée au problème. Cette heuristique consiste à faire rentrer les produits au plus tôt, tout en évitant les conflits, jusqu'à l'apparition d'un cycle.

Elle consiste à minimiser une fonction de coût en respectant des contraintes, le critère et les contraintes étant des fonctions linéaires des variables du problème [Mellouli et al, 04].

- ***Techniques de satisfaction de contraintes***

Le problème de satisfaction de contraintes connu sous le nom "Constraints Satisfaction Problem" (CSP) a longtemps suscité l'intérêt dans le domaine de l'intelligence artificielle, et de plus en plus, les techniques de satisfaction de contraintes ont été étudiées en tant que moyen pour résoudre les problèmes d'ordonnement [Minton et al, 92] [Smith et al, 93] [Cheng et al, 94].

La formulation des algorithmes CSP diffère selon le type et le niveau de complexité du problème posé.

Il y a différentes façons de formuler ce problème comme un CSP. Le plus souvent, il a été formulé comme un problème de trouver un ensemble de dates début pour chaque opération des produits introduits dans la ligne de production.

Les méthodes de résolution de CSP permettent aussi la réduction de l'espace de recherche à explorer par l'exploitation des contraintes du problème. C'est ainsi que certaines solutions ne seront pas évaluées puisque l'étude des contraintes du problème indique qu'elles sont impossibles.

Parmi les problèmes de CSP les plus connus figurent le problème des reines et le problème de coloration de graphe. Le problème des reines consiste à placer n reines sur n différentes cases d'un échiquier de n lignes par n colonnes de manière à ce qu'aucune reine ne puisse en éliminer une autre [Tsang, 99]. Une des principales particularités de ce problème est que le nombre de reines détermine à la fois le nombre de variables du problème et la taille du domaine de celles-ci.

Le problème de coloration de graphe est un problème où l'on doit affecter une couleur à chacune des zones d'une carte de manière à ce qu'aucune zone adjacente de cette dernière ne soit de la même couleur [Zhou et al, 1999].

Enfin, le problème du HSP est un autre problème retrouvé dans la littérature initié avec les travaux de Yih et Cheng [Yih, 94], [Cheng et al, 97].

Par la suite, Hindi et Fleszar [Hindi et al, 04] ont travaillé sur l'amélioration des résultats obtenus par Yih et Cheng. D'autres recherches ont été également réalisées dans cette optique [Brailsford et al, 99].

La méthode de résolution exploitée dans ce mémoire, est détaillée dans le chapitre II.

- ***Branch and Bound***

L'algorithme branch and bound permet d'obtenir le 1-cycle optimal pour le problème cyclique mono-produit [Chen et al, 94], [Chen et al, 95]. La borne utilisée dans l'algorithme est, dans ce cas, calculée grâce à un programme linéaire.

Dans le cas cyclique multi-produits, Varnier et Jeunhomme [Varnier et al, 00] ont développé une méthode de type branch and bound qui parcourt un arbre dans lequel toutes les combinaisons sont proposées.

- ***Algorithmes génétiques***

Les algorithmes génétiques sont des algorithmes itératifs dont le but est d'optimiser une fonction prédéfinie, appelée fonction fitness [Holland, 75], [Goldberg, 89]. L'utilisation de ces algorithmes, proposée par [Lim, 97] permet de trouver un cycle proche de l'optimum à partir d'un ensemble de solutions existantes (population initiale). Pour chaque solution, on crée une force (fitness) qui reflète sa qualité, et un sous-ensemble de solutions est sélectionné (en fonction de leur force) pour se reproduire. La descendance est obtenue avec des opérateurs de croisement et de mutation ; certains parents sont éliminés (en fonction de la fonction à optimiser) pour être remplacés par les enfants. On recommence ceci jusqu'à la fin de l'itération ou jusqu'à ce qu'aucun enfant ne soit meilleur que les parents. Cette méthode sera détaillée dans le second chapitre de ce mémoire.

1.4.2. Généralisation

D'autres types de travaux sont effectués sur le HSP comme ceux relatif aux problèmes d'implantation de lignes [Grunder et al, 97], la création de modèles objet [Goujon et al, 97], ou

graphiques type réseaux de Petri P-temporels [Khansa, 97]. Ces recherches s'intéressent à la commande du système d'un point de vue robustesse [Collart-Dutilleul, 97]. Enfin, les problèmes environnementaux commencent à être pris en compte lors de l'ordonnancement des ateliers [Subai et al, 03]. Le problème consiste à partir des solutions de l'ordonnancement réalisables et de déterminer celle qui est la mieux adaptée aux contraintes environnementales.

1.5. Position du problème

Le problème d'ordonnancement abordé dans le cadre de ce mémoire est un problème de type job shop et plus particulièrement SHMP avec ressources disjonctives, prenant en compte les contraintes temporelles, les contraintes de disponibilité des ressources, les contraintes de disponibilité du robot et la variation des temps des transferts du robot.

Le problème job shop est NP-difficile [Lenstra et al, 77] et donc également le problème SHMP étudié puisqu'il dérive d'un problème job shop.

Un ensemble de tâches appartenant à des projets, doit être exécuté, les bornes temporelles représentent respectivement les marges minimales et maximales d'une tâche, le temps d'exécution étant inclus dans cet intervalle.

Une tâche peut impliquer plusieurs ressources. Un ordonnancement est dit faisable, si toutes les contraintes de précédence sont respectées et si pour chaque ressource, une tâche au plus lui est affectée à un instant donné.

Étant donné le problème d'optimisation du critère temps du cycle, nous envisageons d'adapter une approche de résolution comportant une heuristique et une metaheuristique.

La première consiste à développer un algorithme de satisfaction de contraintes introduisant les temps de déplacements du chariot, considérés négligeables ou constants dans la majorité des travaux traitant ce problème [Yih, 94], [Cheng et al, 97], [Hindi, 04]; une hybridation est ensuite envisagée pour améliorer les résultats ainsi trouvés.

La deuxième est relative à l'algorithme génétique d'un grand intérêt pour la résolution des problèmes d'ordonnancement.

Une fois l'approche résolution est établie, une évaluation de la robustesse de l'ordonnancement est, par la suite, menée par l'introduction de nouvelles mesures de robustesse permettant la mesure de la performance de l'ordonnancement dès la phase initiale de génération des ordonnancements.

1.6. Conclusion

Dans ce chapitre, après l'introduction des principales caractéristiques des lignes de traitement de surfaces ; nous nous sommes intéressés, d'abord, à la complexité des problèmes qui peuvent être rencontrés et ensuite à leur résolution, par différentes méthodes de exactes et approchées envisageables. Les approches exactes, bien qu'elles soient assez nombreuses et diversifiées, demeurent insuffisantes et peu adaptées aux divers types de problèmes d'ordonnancement.

Les méthodes approchées exploitant des métaheuristiques hybridées ou pas sont adaptées. Pour obtenir une bonne solution au problème d'ordonnancement de l'atelier étudié, les contraintes spécifiques à ce type d'industrie, à savoir : les ressources, le chariot, les produits, ainsi que les critères à minimiser, sont à préciser.

Dans le chapitre suivant, un algorithme de satisfaction de contraintes est développé pour trouver au moins une solution faisable, exploité en deuxième phase par hybridation pour l'obtention de meilleurs résultats.

Chapitre II : Méthodes de résolution du problème SHMP illustré par une ligne de traitement de surfaces

II.1. Introduction

Les problèmes job shop, étant classés parmi les problèmes NP-difficiles, sont complexes à résoudre et en particulier pour le cas du "Hoist Scheduling Problem" HSP ; différents types de HSP peuvent être distingués [Hindi, 04]:

- Mono-robot /Mono-produit, Single-Hoist/Single-Product
- Mono-robot /Multi-produits, Single-Hoist/Multi-Products
- Multi-robots/ Mono-produit, Multi-Hoists/Single-Product
- Multi-robots/ Multi-produits. Multi-Hoists/Multi-Products

La complexité réside dans le fait qu'il n'y a qu'un seul et unique robot chargé d'effectuer tous les déplacements; ce problème est connu sous le nom "Single Hoist Scheduling" SHS [Lei et al,94], [Lim, 97], [Zhou et al, 03], [Subai et al,06]. Ce problème devient beaucoup plus contraint lorsqu'on a plusieurs produits à traiter avec un seul et unique robot, et est connu sous le nom "Single Hoist Multiple Products Scheduling" SHMPS [Yih,94], [Cheng et al, 97], [Hindi et al, 04]. Dans le domaine de galvanoplastie [Chauvet et al, 00], [Subai et al, 06], c'est, par exemple, le cas de l'atelier de traitement de surfaces, étudié, dans ce mémoire, vient concrétiser le type du problème Mono-robot /Multi-produits, [Mhedhbi et al, 08^a] [Mhedhbi et al, 08^b], [Mhedhbi et al, 08^c], [Mhedhbi et al, 10], [Mhedhbi et al, 11].

Deux approches peuvent être considérées pour résoudre ce genre de problèmes : les méthodes exactes et les méthodes approchées.

Parmi les méthodes exactes, on peut citer essentiellement les méthodes de programmation linéaire [Song et al, 97] les méthodes de type Branch & Bound [Chen et al, 94], [Chen et al, 95], [Varnier et al, 00] et les algorithmes de satisfaction de contraintes [Yih, 94] [Varnier, 96], [Cheng et al, 97], [Hindi, 04]. Les algorithmes génétiques et les essaims particuliers constituent des méta-heuristiques généralement adoptées pour résoudre les problèmes complexes [Lim, 97],

[Pezzella et al, 2008], [Boukef et al, 09] et obtenir des solutions meilleurs sinon optimales.

Au cours des dernières années, de nombreux travaux ont porté sur l'hybridation entre méta-heuristiques ou heuristiques avec les approches exactes [Jourdan et al, 2008].

Ces stratégies donnent généralement de bons résultats, parce qu'ils sont capables d'exploiter simultanément les deux types de méthodes.

Après la description détaillée de l'atelier de traitement de surfaces étudié, une modélisation globale est donnée dans ce chapitre par les gammes logiques suivie d'une modélisation plus détaillée donnée par les gammes opératoires restreintes.

L'approche proposée pour résoudre le problème job shop mono-robot/multi-produits relatif à l'atelier de traitement de surfaces passe d'abord par la recherche d'une première solution qu'il s'agit d'améliorer par les algorithmes génétiques. Elle consiste ensuite à développer un algorithme de satisfaction de contraintes basé sur le calcul des temps d'attente du robot et intégrant essentiellement les temps de déplacement de cette ressource considérés négligeables ou constants dans la majorité des travaux existants.

En effet, ce problème considère un ensemble d'opérations associées à leurs domaines temporels et un ensemble de déplacements du robot, le tout géré par un ensemble de contraintes temporelles et de contraintes d'antériorité.

La solution est alors un ordonnancement attribuant une valeur pour chaque opération et présentant l'ordre des déplacements du chariot, de sorte que toutes les contraintes sont satisfaites [Mhedhbi et al, 11].

Une hybridation entre cette approche proposée et deux heuristiques classiques (Shortest Processing Time First/ SPTF, Longest Processing Time First / LPTF) [Pinedo, 05], est, ensuite proposée dans le but d'améliorer les résultats obtenus par l'application de l'algorithme de satisfaction de contraintes [Mhedhbi et al,10].

L'optimisation de ces résultats est ensuite considérée par application des algorithmes génétiques [Mhedhbi et al, 11^a], la génération de la population initiale étant obtenue en appliquant l'algorithme de satisfaction de contraintes hybridé avec SPTF.

Les résultats obtenus pour l'atelier de traitement de surfaces étudié sont présentés et discutés dans

ce chapitre.

II.2. Description de l'atelier de traitement de surfaces étudié

Le traitement de surfaces consiste à déposer sur la surface d'une pièce métallique un film de protection très fin au moyen d'un dépôt chimique ou électrolytique. Pour déposer ce film, là où les pièces sont plongées dans une succession de cuves contenant de solutions chimiques, on a recours à des robots de levage.

La ligne de traitement de surfaces considérée, est un problème typique du problème SHMP conçue pour traiter une variété de pièces de nickelage, de zingage et d'argentage. En effet, cette ligne est composée de 23 machines qui sont visitées, grâce à un chariot dans un ordre qui dépend des gammes à effectuer. Le pilotage consiste à déterminer la séquence des mouvements du robot de levage de la ligne.

II.2.1. Contexte de production

-Chaque cuve est mono-bac, donc ne peut recevoir qu'un produit à la fois ;

-Les postes de chargement et déchargement sont confondus ;

Notre système peut comporter plusieurs produits et chaque produit ayant un ordre de passage différent dans les cuves avec des intervalles de trempe variables. Comme nous traiterons le cas multi-produits, des hypothèses supplémentaires doivent être prises en compte :

-l'atelier étudié est relatif à un problème cyclique, assurant un ratio de production égal à $1/3, 1/3, 1/3$, dans lequel ne nous tenons pas compte du régime transitoire, [Korbaa, 98], [Korbaa, 03], qui constitue par lui-même un problème qu'il est important de résoudre ;

-le robot déplace le produit d'une cuve T_i vers une cuve T_j (T_j pouvant être différent de T_{i+1} , les gammes étant différentes) ;

-la durée de transfert d'un produit d'une cuve à une autre doit tenir compte de la durée d'égouttage du produit.

L'objectif est donc de trouver un ordonnancement qui satisfait les diverses contraintes suivantes :

➤ **Contrainte de fenêtre de temps**

La durée opératoire de chaque traitement dans une cuve diffère d'un produit à l'autre. Cette durée est comprise entre une marge minimale et une marge maximale. Si la borne minimale est non atteinte ou la borne maximale est dépassée, le produit risque d'être détérioré.

➤ **Contrainte de capacité des cuves**

La contrainte de capacité unitaire des cuves exige que quand un produit j arrive à une cuve T_i , celle-ci doit être vide, autrement dit, le produit occupant cette cuve doit avoir été évacué au préalable.

➤ **Contrainte de capacité du robot**

Comme le robot ne peut transporter qu'un seul tonneau à la fois, il ne doit jamais exister de conflit concernant l'utilisation du robot entre les différents produits.

➤ **Contrainte d'antériorité**

C'est une contrainte de succession entre deux tâches d'une même gamme.

➤ **Contrainte de non préemption**

Une fois commencée, une tâche ne peut être interrompue. On ne pourra avoir recours à une segmentation de l'exécution d'une tâche.

➤ **Contrainte disjonctive**

Un travail ne peut être effectué que sur une seule cuve à la fois, l'exécution des tâches d'un même produit doit ainsi être séquentielle et non parallèle. De même, une machine ne peut traiter qu'une tâche à la fois.

II.2.2. Les ressources du système

Le passage d'une cuve à une autre s'effectue à l'aide de différents systèmes de manipulation :

➤ le chariot : c'est un translateur de tonneau permettant le déplacement d'une cuve à une autre. Il est capable d'effectuer 4 mouvements : la montée et la descente commandées par un moteur frein de levage à deux sens de rotation, le déplacement à gauche et à droite commandé par un moteur frein d'entraînement à deux vitesses et à deux sens de rotation.

Le chariot se déplace avec 2 deux vitesses, la plus petite est pour le début et la fin du déplacement entre 2 bains. Par exemple, si le produit doit être transporté de T_k vers T_j , avec $j > k$, la vitesse du chariot de T_k vers T_{k+1} et de T_{j-1} vers T_j est de 0,15 m/s. Pour le reste des mouvements, ça sera 0,57 m/s. Pour simplifier, considérons la moyenne 0,36 m/s. de deux vitesses pour tous les mouvements du chariot. Les déplacements élémentaires sont calculés et donnés dans le tableau II-1. On suppose qu'il n'y a pas de différence entre le chariot en charge ou à vide. C'est le cas lorsque le poids du produit est négligeable par rapport au poids du tonneau. A noter qu'un déplacement entre 2 cuves non succesives est la somme des déplacements élémentaires.

Tableau II-1 Déplacements élémentaires du chariot

θ_{jk}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0																						
2	1.9	0																					
3		1.9	0																				
4			1.9	0																			
5				1.8	0																		
6					1.95	0																	
7						1.8	0																
8							1.7	0															
9								1.7	0														
10									1.8	0													
11										1.8	0												
12											1.7	0											
13												1.8	0										
14													1.8	0									
15														1.7	0								
16															1.85	0							
17																1.85	0						
18																	1.7	0					
19																		1.7	0				
20																			1.7	0			
21																				1.7	0		
22																					1.7	0	
23																						1.75	0

θ_{jk} est le temps de déplacement du chariot de la cuve T_j vers T_k , $0 \leq j < N$, $0 < k \leq N$

➤ le tonneau : c'est un récipient hexagonal fermé et mis en rotation, il est doté d'un moteur de rotation qui permet de supporter une charge maximale de 30 Kg; la chaîne de traitement de surfaces est équipée de sept tonneaux répartis comme suit :

- 4 tonneaux pour le cycle de zingage,
- 2 tonneaux pour le cycle de nickelage,
- 1 tonneau pour le cycle d'argentage.

La limitation du nombre de tonneaux oblige à travailler sur des cycles de production 4, 2, 1 pour assurer une production maximale.

➤ les cuves de traitement : elles servent à recevoir les produits nécessaires aux traitements souhaités. Le contenu des cuves est régulièrement vérifié afin de maintenir un niveau et une qualité acceptables pour l'utilisation requise.

Chacune de ces cuves est caractérisée par le type de traitement qu'elle effectue, et par sa position sur la ligne.

Le tableau II-2, ci-dessous, présente les différentes opérations de traitement de surfaces pour chaque cuve.

Tableau II-2-Détails des différentes cuves de la ligne de traitement de surfaces

N° cuve	Opération	Fenêtre de temps
T_1	Dégraissage chimique	[300, 600]
T_2	Rinçage	[60, ∞[
T_3	Décapage acide	[600, 1800]
T_4	Rinçage	[60, ∞[
T_5	Dégraissage électrolytique	[60, 180]
T_6	Dégraissage électrolytique	[60, 180]
T_7	Rinçage	[60, ∞[
T_8	Activation	[60, 300]
T_9	Rinçage	[60, ∞[
T_{10}	Nickelage	[1800, 1800]
T_{11}	Récupération Nickel	[60, 600]
T_{12}	Rinçage	[60, ∞[
T_{13}	Argentage	[180, 300]
T_{14}	Zingage	[1800, 1800]
T_{15}	Rinçage	[60, ∞[
T_{16}	Zingage	[1800, 1800]
T_{17}	Rinçage	[60, ∞[
T_{18}	Passivation Bleue	[60, 60]
T_{19}	Passivation olive	[60, 90]
T_{20}	Passivation tropicale	[60, 60]
T_{21}	Rinçage	[60, ∞[
T_{22}	Rinçage	[60, ∞[
T_{23}	Chargement et Déchargement	[60, ∞[

II.2.3. Les opérations de traitement

Les opérations de traitement sont au nombre de 11 :

- dégraissage (cuve 1) : élimination des huiles et graisses se trouvant à la surface des pièces destinées à recevoir un revêtement métallique, soit aux solvants organiques intervenants en phase liquide ou en phase vapeur, soit aux lessives alcalines très chaudes soit dans des cuves alcalines par électrolyse,
- décapage (cuve 3) : élimination des couches superficielles de métal amorphe ou obtention d'une surface rugueuse par voie chimique, électrolytique ou mécanique,
- attaque anodique (cuve 5 ,6) : attaque électrolytique d'une surface métallique en cuve acide ou alcalin, la pièce étant placée à l'anode, en vue de préparer la surface pour obtenir l'adhérence d'un dépôt électrolytique ultérieur, (le terme décapage est souvent employé dans le même sens),
- nickelage (cuve 10) : dépôt électrolytique de nickel fait soit directement sur fer et acier soit sur sous-couches de cuivre, dans un but de décoration ou de protection contre la corrosion,
- argentage (cuve 13) : revêtement d'argent sur d'autres métaux en vue de leur procurer la brillance et l'inaltérabilité de ce métal (sens général), le dépôt électrolytique d'argent soit dans un but décoratif utilisé en orfèvrerie, soit dans un but de protection (sens particulier d'usage courant),
- zingage (cuve 14, 16) : revêtement de zinc sur un autre métal, quelque soit le procédé (sens général) utilisé couramment dans le sens de zingage électrolytique : dépôt électrolytique de zinc (sens particulier d'usage courant),
- rinçage (cuve 2, 4, 7, 9, 12, 15, 21, 17, 22) : opération consistant à faire des trempages successifs dans l'eau courante ou des aspersion au jet d'eau sous pression en vue d'enlever les produits chimiques restant à la surface d'une pièce métallique à la sortie d'une cuve chimique,
- passivation bleue, tropicale et olive (cuve 18, 19, 20), c'est une technique de rendre (un métal) insensible à la corrosion par un traitement déterminé,
- charge et décharge (cuve 23),
- activation (cuve 8),
- récupération Nickel (cuve 11).

II.2.4. *Temporisation par cycle*

Vu que le nombre d'articles à traiter est important, il a été judicieux de répartir les articles en familles et présenter le cahier des charges d'un produit par famille, tableaux II-3, II-4 et II-5

Tableau II-3-Cycle de zingage

Opération	Traitement chimique	Cuve possible pour traiter cette opération	Fenêtre de temps (seconde)
O_{11}	Dégraissage chimique	T_1	[300, 600]
O_{12}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{13}	Décapage acide	T_3	[600, 1800]
O_{14}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{15}	Dégraissage électrolytique	$T_5 T_6$	[60,180]
O_{16}	Dégraissage électrolytique	$T_5 T_6$	[60,180]
O_{17}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{18}	Zingage	$T_{14} T_{16}$	[1800, 1800]
O_{19}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{110}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{111}	Passivation bleue	T_{18}	[60, 60]
O_{112}	Passivation olive	T_{19}	[60, 90]
O_{113}	Passivation tropicale	T_{20}	[60, 60]
O_{114}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{115}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{116}	Chargement et Déchargement	T_{23}	[60, ∞ [

Tableau II-4-Cycle de nickelage

Opération	Traitement chimique	Cuve possible pour traiter cette opération	Fenêtre de temps (seconde)
O_{21}	Dégraissage chimique	T_1	[300, 600]
O_{22}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{23}	Dégraissage électrolytique	$T_5 T_6$	[60, 180]
O_{24}	Dégraissage électrolytique	$T_5 T_6$	[60, 180]
O_{25}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{26}	Activation	T_8	[60,300]
O_{27}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{28}	Nickelage	T_{10}	[1800, 1800]
O_{29}	Récupération Nickel	T_{11}	[60, 600]
O_{210}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{211}	Chargement et Déchargement	T_{23}	[60, ∞ [

Tableau II-5-Cycle d'argentage

Opération	Traitement chimique	Cuve possible pour traiter cette opération	Fenêtre de temps (seconde)
O_{31}	Dégraissage chimique	T_1	[300, 600]
O_{32}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{33}	Dégraissage electrolytique	$T_5 T_6$	[60, 180]
O_{34}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{35}	Activation	T_8	[60, ∞ [
O_{36}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60,300]
O_{37}	Argentage	T_{13}	[180, 300]
O_{38}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{39}	Rinçage	$T_2 T_4 T_7 T_9 T_{12} T_{15} T_{17} T_{21} T_{22}$	[60, ∞ [
O_{310}	Chargement et Déchargement	T_{23}	[60, ∞ [

II.3. Modélisation par RdPT

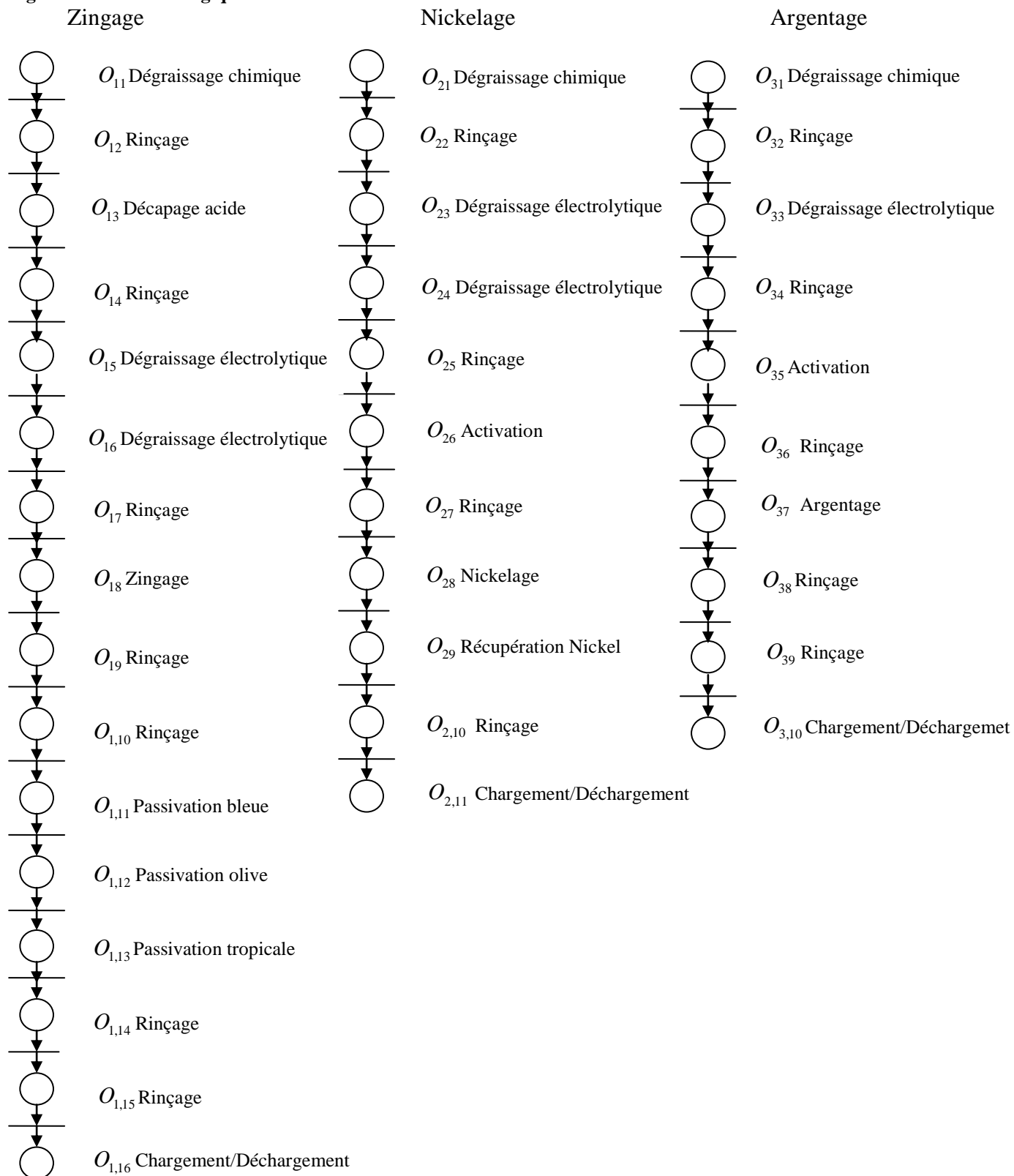
Le développement du modèle réseau de Petri temporel, RdPT, sur lequel repose la modélisation envisagée est basé sur la connaissance des gammes de fabrication; nous nous appuyons ici sur les modèles présentés dans [OhL, 95], [Camus, 97].

Nous allons modéliser l'entrée en ligne d'une séquence de gammes chacune présentant un type de produits. Toute la difficulté réside dans la gestion simultanée de plusieurs gammes avec des spécificités distinctes. Quelque soit l'outil de modélisation adapté, le cas multi-produits peu difficilement être modélisée. Cet handicap est dû essentiellement à l'explosion combinatoire engendrée par la multitude des gammes en présence. Nous allons donc représenter une seule gamme, celle d'argentage pour exposer clairement le niveau de complexité du problème posé. Pour la modélisation, nous avons eu recours aux Gammes Logiques GL pour présenter l'ensemble des gammes suivies des Gammes Opératoires Restreintes Temporisées GORT pour présenter la gamme d'argentage [Camus, 97], [Amar, 94].

II.3.1. Gammes logiques

Une *gamme logique* (GL) spécifie les opérations et l'ordre dans lequel celles-ci doivent être exécutées. Les GLs ne spécifient donc ni les transferts, ni les ressources qui effectuent les opérations. La figure II-1 présente les gammes logiques relatives au zingage, au nickelage ou à l'argentage.

Figure II-1-Gammes logiques associées à l'atelier de traitement de surfaces étudié



II.3.2. Gammes opératoires restreintes

L'étape considérée ici consiste à ajouter aux GLs les informations relatives au procédé afin de spécifier sur quelle ressource une opération est effectuée. Le résultat est une Gamme Opératoire (GO). Comme le montrent les tableaux II-3, II-4 et II-5, une même opération peut s'effectuer sur plusieurs ressources, nous parlons alors de flexibilité d'affectation et nous introduisons pour cela la notion de Gamme Opératoire Restreinte (GOR) dans laquelle toute opération est associée à différentes ressources spécifiées pour le même traitement.

En complétant les GORs par le remplacement de chaque transition représentative d'une opération de transfert par l'élément transition immédiate / place / transition temporisée. Nous définissons ainsi les Gammes Opératoires Restreintes Temporisées (GORT).

Les transferts entre les lieux physiques où les opérations sont exécutées ne peuvent pas être détaillés tant que les opérations restent potentiellement affectées à plusieurs ressources et non pas à une ressource particulière (suite à l'ordonnancement).

A ce stade du développement du modèle, il faut alors décider si les transferts entre les lieux physiques peuvent être négligés ou si des estimations des durées peuvent être incluses dans le modèle. Pour s'approcher plus des problèmes industriels réels, on ne peut pas négliger les transferts vu que pour certains ateliers tels que le cas du traitement des surfaces, le temps mis pour certains déplacements est proche de certains temps de traitement. Nous affectons alors à ce stade une valeur moyenne ($T_M=47\text{sec}$) pour le temps de déplacement qui sera calculé en détail lors du développement de l'algorithme, une fois une ressource particulière sélectionnée.

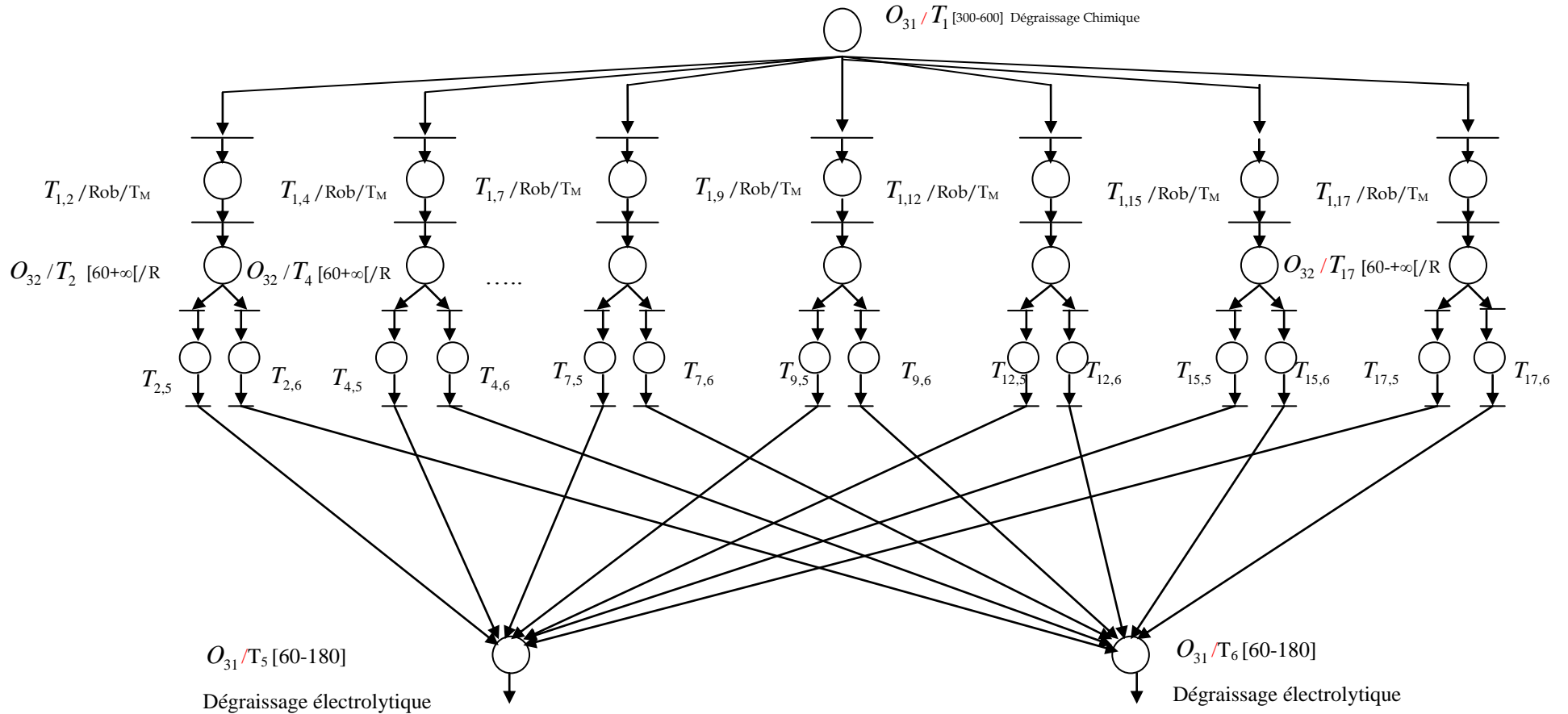
Cette dualité entre ressources et gammes est parfaitement mise en évidence avec le modèle RDPT présentant la gamme de l'argentage dans la figure II-2. Pour ne pas encombrer la figure, on a travaillé avec 7 bains de rinçage et non 9 bains comme indiqué dans le tableau II-5.

Les gammes opératoires restreintes temporisées relatives au nickelage et au zingage sont modélisées de la même façon que la gamme d'argentage définie dans la figure II-2.

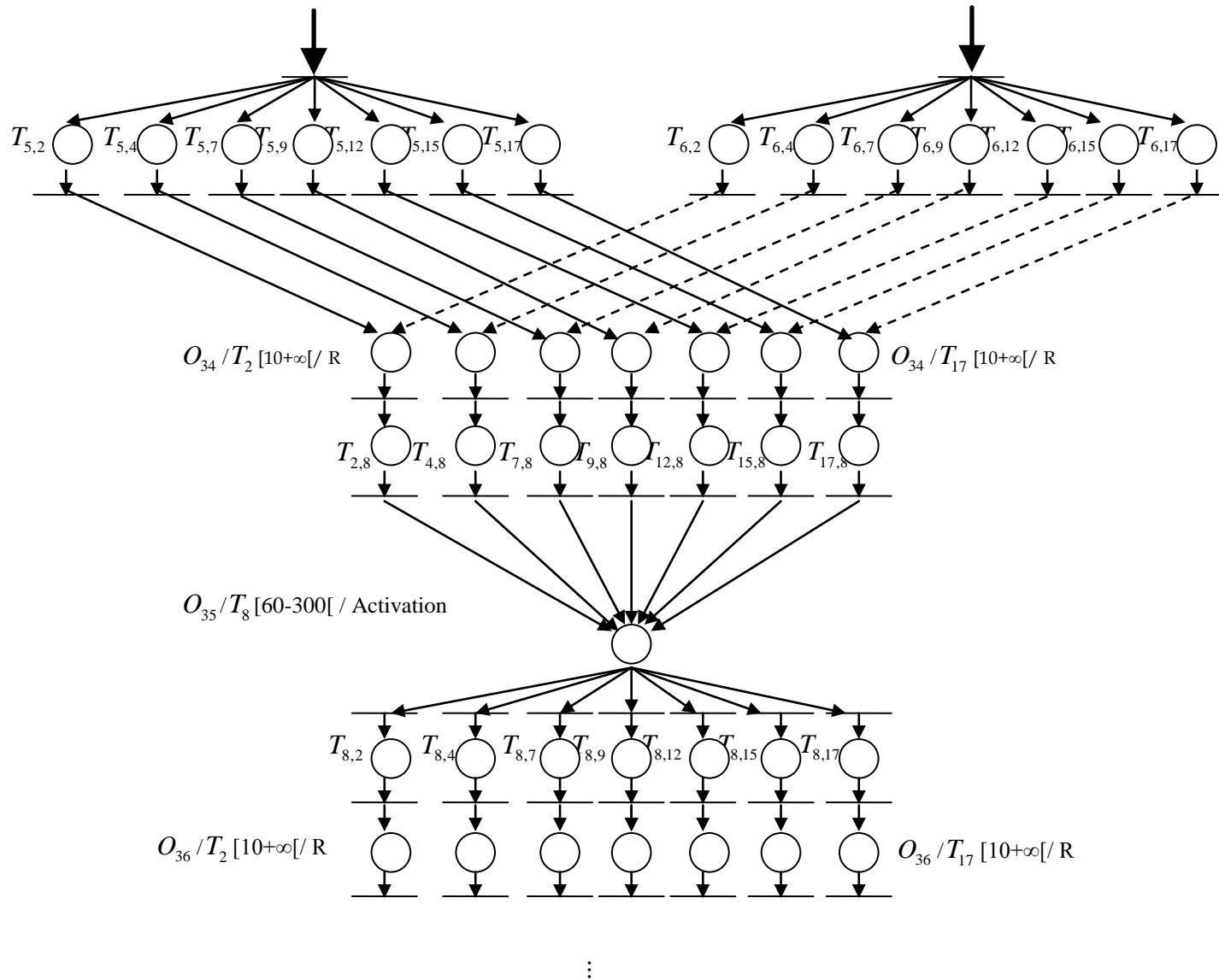
La figure II-3 montre l'aspect cyclique du système étudié ainsi que les ratios de production considérés, (1/3, 1/3, 1/3), dans les simulations.

Rob : Robot

R : Rinçage



$T_{i,j}$: Déplacement de la cuve i vers la cuve j



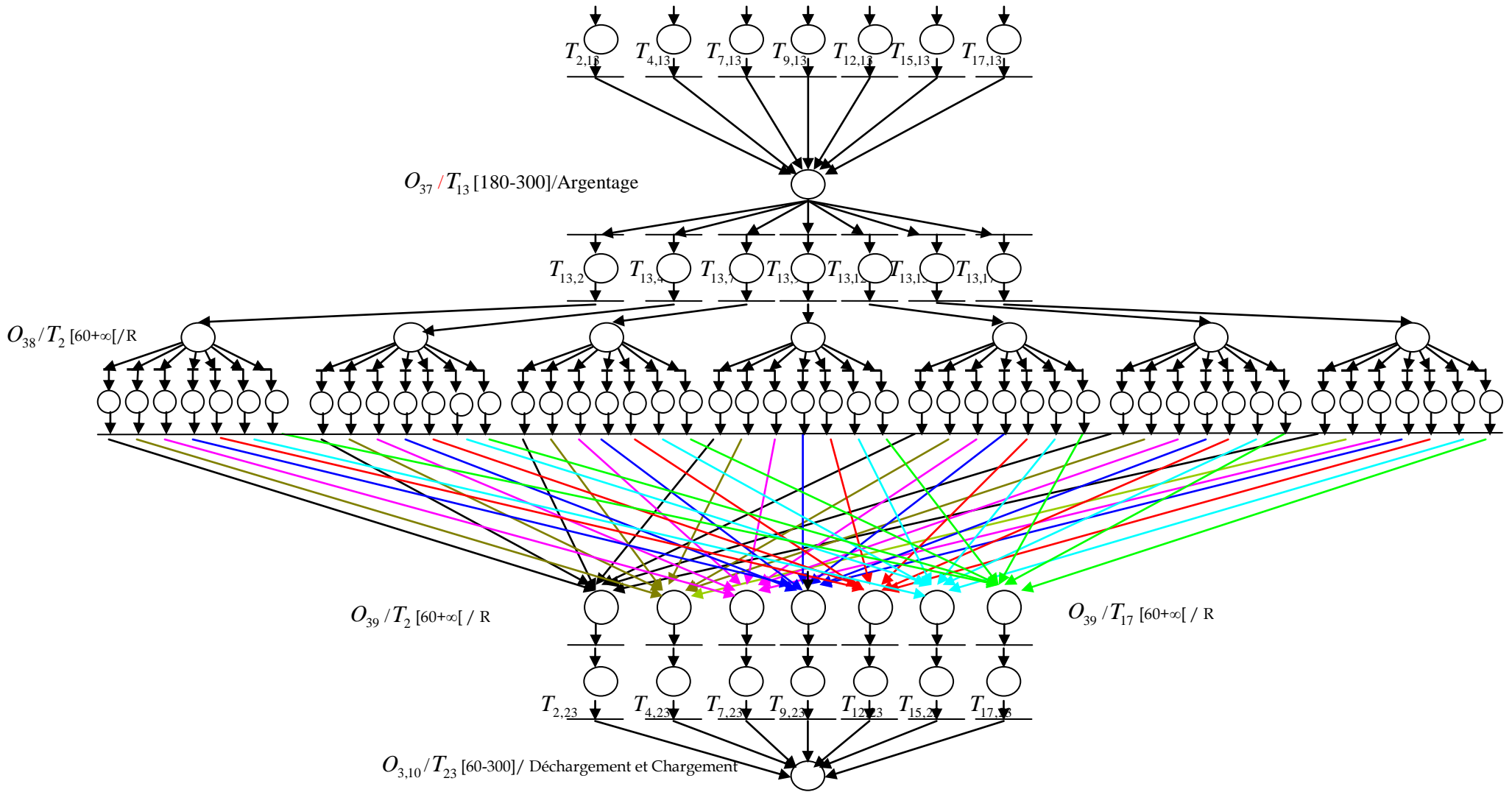


Figure II-2-Gamme opératoire restreinte temporisée, Argentage

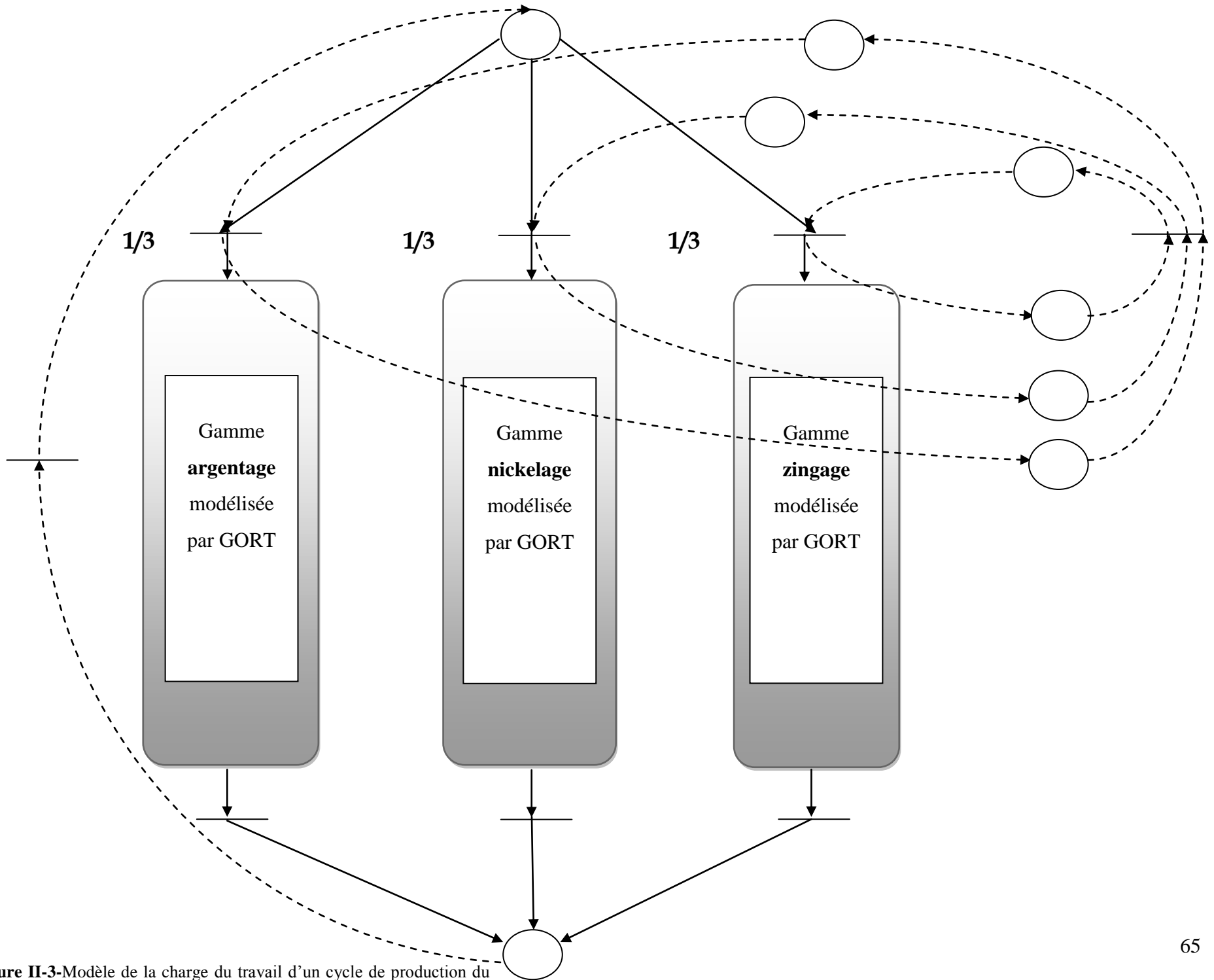


Figure II-3-Modèle de la charge du travail d'un cycle de production du régime permanent sur un horizon de production discret minimal (=3)

Une modélisation de la gamme d'argantage par réseaux de Petri tenant compte des ressources et des déplacements du chariot met en relief la complexité combinatoire du problème traité : il faut tenir compte également de l'aspect concurrent aux mêmes ressources de plusieurs gammes et enfin le transport avec le robot. Une recherche d'un ordonnancement combinant 3 gammes similaires à celle de l'argantage nécessite le développement d'un algorithme tenant compte de l'ensemble des contraintes et pouvant; l'algorithme de satisfaction de contraintes constitue une approche à cette complexité.

II.4. Algorithme de Satisfaction de Contraintes (ASC)

Comme il a été précédemment rappelé, l'atelier étudié dans ce mémoire est un problème typique du SHS, et en particulier du SHMPS.

En effet, nous étudions en particulier le problème d'ordonnancement multi-produits avec un seul chariot et multiples cuves (cuves similaires). Dans un problème multi-produits, r pièces sont introduites dans le système (pour la conservation de flux, r pièces sont retirées de la ligne) toutes les périodes. La ressource critique pour ce type du problème est le chariot. L'objectif de cette étude est de développer un algorithme de satisfaction de contraintes afin de construire d'abord l'ordonnancement des mouvements du chariot et en deuxième phase d'optimiser cet ordonnancement en minimisant le temps de cycle.

Un problème de satisfaction de contraintes consiste en un ensemble de variables et un ensemble de contraintes. Une solution est une combinaison spécifiant une valeur pour chaque variable, de sorte que toutes les contraintes soient satisfaites. Une méthode de satisfaction de contraintes prend les variables et les contraintes et commence par générer un assignement initial pour les variables.

Le séquençement initial est alors régénéré à plusieurs reprises jusqu'à ce qu'une solution soit trouvée. Afin d'optimiser le temps de cycle, l'algorithme de satisfaction de contraintes est compilé plusieurs fois pour réduire les temps d'exécution par itération jusqu'à atteindre une valeur stagnée de temps du cycle.

Un problème de satisfaction de contraintes est un triplet $P = \{X, D, C\}$, X étant un ensemble $\{X_1, X_2, \dots, X_n\}$ de variables, D un ensemble $\{D_1, D_2, \dots, D_n\}$ de domaines tels que D_i est le

domaine de valeurs pouvant être prises par X_i et C un ensemble de contraintes restreignant les valeurs $v \in D_i$ pouvant être prises par X_i , [Dechter et Frost, 02].

II.4.1. Travaux antérieurs

Avant d'exposer l'approche développée pour résoudre le problème SHMP, il est important de présenter les recherches antérieures traitant le même type problème d'ordonnement.

La plupart des recherches existantes traitent le problème le moins complexe qui est le mono-produit/mono-robot (SHSP), dont l'objectif est de trouver le temps de cycle minimum.

En revanche, peu de travaux considèrent le problème plus complexe qui est le multi-produits/mono-robot (SHMP), qui fait l'objet de ce travail [Yih, 94], [Cheng et al, 97], [Brailsford et al, 99], [Hindi, 04].

Les techniques de satisfaction de contraintes et les modèles heuristiques ont été étudiées en tant que moyen pour résoudre les problèmes d'ordonnement. Pour résoudre ce problème, il y a différentes façons de formuler ce problème comme un problème de satisfaction de contraintes. Le plus souvent, il a été formulé comme un problème d'exploiter la capacité des ressources.

II.4.1.1. Algorithmes de Yih

Deux heuristiques ont été développées dans [Yih, 94], la première présente une planification des produits, compte tenu des délais de traitement minimums et sans introduire les conflits du chariot dans un premier stade; le deuxième algorithme améliore les résultats précédemment trouvés en introduisant les conflits au niveau du chariot mais en supposant que le temps de déplacement est constant.

II.4.1.2. Algorithmes de Hindi

Dans [Hindi, 04] est développé un algorithme non standard de satisfaction de contraintes pour résoudre le problème SHMP, où les variables ont des étiquettes et l'ordre de l'étiquetage indique l'ordre des opérations de levage, qui sont choisies et programmées en fonction de leur début au plus tôt. Cet algorithme produit des solutions beaucoup mieux que celle de Yih (1994) mais tout en imposant constants les temps de déplacement du chariot.

II.4.2. Formulation du problème

Notations

N	: nombre de cuves $T_k = \{T_1, \dots, T_N\}$, $k = 1, \dots, N$
T_k	: $k^{\text{ème}}$ cuve pour $k = 1, \dots, N$
N_k	: nombre de cuves multiples $k = 1, \dots, N$
J	: nombre de produits
M	: nombre d'opérations
M_j	: nombre d'opérations du $j^{\text{ème}}$ produit, $j = 1, \dots, J$
O_{ji}	: $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit
a_{ji}^k, b_{ji}^k	: borne minimale et maximale du temps de traitement pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve
P_{ji}^k, α_{ji}^k	: actuel et nouveau temps de traitement pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve
θ_{jk}	: temps de déplacement du chariot de la cuve T_j vers T_k , $0 \leq j < N$, $0 < k \leq N$
δ_{ji}^k	: temps d'attente du chariot pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit au niveau de la $k^{\text{ème}}$ cuve
S_j	: séquencement des opérations pour le $j^{\text{ème}}$ produit
S	: séquencement des opérations de tous les produits
Z_{ji}^k	: charge de la cuve pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve

$Z = \max_{k=1,\dots,N} Z_j^k$: maximum des charges des cuves pour $k = 1, \dots, N$

$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$: charge totale des cuves pour $k = 1, \dots, N$, $i = 1, \dots, M_j$, $j = 1, \dots, J$

$X = \sum_{j=1}^J \sum_{i=1}^{M_j} \theta_{ji}$: temps total des déplacements du chariot pour $i = 1, \dots, M_j$, $j = 1, \dots, J$

$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$: temps total d'attente du chariot pour $k = 1, \dots, N$, $i = 1, \dots, M_j$, $j = 1, \dots, J$

$Y = X + W$: charge du chariot

$CT = \max(Y, Z)$: temps de cycle

CT_{\min} : temps du cycle minimal

R : nombre d'implémentations de l'algorithme

S^r : ordonnancement relatif à la $r^{\text{ème}}$ implémentation pour $r = 1 \dots R$

Un ordonnancement est réalisable s'il satisfait les contraintes suivantes.

1. Contrainte de capacité de cuves : quand un produit atteint la cuve destination, celle-ci doit être vide. Une cuve traitant un produit ne peut être disponible qu'après l'écoulement du temps de traitement associé à ce produit,

2. Contrainte de fenêtre de temps : le temps de traitement d'un produit a une borne minimale et une borne maximale à respecter impérativement et pour certains traitements la borne supérieure est infinie,

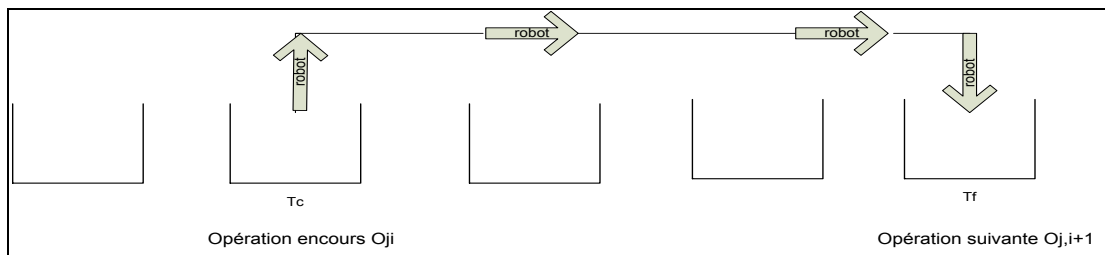
3. Contrainte de capacité du chariot : le chariot doit avoir suffisamment de temps pour effectuer les déplacements entre les cuves.

Notons par $Indicateur = [I_1, I_2, \dots, I_N]$ le vecteur d'occupation des cuves donnant une photo instantanée de l'état d'occupation de ces dernières, $I_k = 1$ quand la cuve est occupée et $I_k = 0$ quand la cuve est vide. I_N est toujours égale à 0 ; en effet, cette cuve de déchargement reçoit les produits qui seront supposés transportés immédiatement vers la zone stockage.

Supposons que le premier déplacement du chariot pour chaque produit commence à partir de la cuve T_N pour aller à la cuve T_j , le temps nécessaire de déplacement pour le chariot sera alors θ_{jN} . Examinons les deux possibilités suivantes pour θ_{jN} :

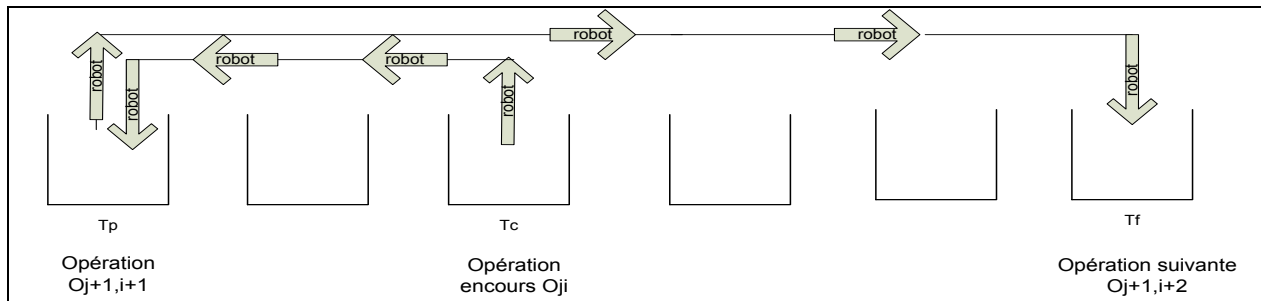
-Si les opérations en cours et la suivante appartiennent au même produit, les cuves T_c et T_f respectivement associés à ces tâches sont sélectionnées et le temps de déplacement du chariot est calculé comme étant la somme des déplacements élémentaires du chariot :

$$\theta_{cf} = \theta_{c,c+1} + \theta_{c+1,c+2} + \dots + \theta_{f-1,f}$$



-Si la prochaine opération et celle en cours n'appartiennent pas au même produit, la dernière opération, appartenant au même produit que celui de la prochaine opération est sélectionnée. Ensuite, les cuves T_p , T_c et T_f respectivement associées à la dernière opération, l'opération encours et la prochaine opération sont sélectionnées et le temps de déplacement du chariot est calculé de la façon suivante :

$$\theta_{cf} = \theta_{c,c+1} + \dots + \theta_{c+k,p} + \theta_{p,p+1} + \dots + \theta_{f-1,f}$$



Afin de respecter les contraintes temporelles, un produit n'est autorisé à rester en dehors de la cuve associée avant l'achèvement de la borne minimale et à l'intérieur après la borne supérieure, le chariot doit être alors disponible au bon moment, selon les deux bornes pour déplacer le produit. Trois cas de disponibilité du chariot peuvent être alors considérés.

-Cas1

Lorsque le chariot est disponible pour déplacer le produit avant l'achèvement de la borne minimale, le chariot doit dans ce cas attendre l'écoulement de la borne inférieure.

-Cas2

Lorsque le chariot est disponible pour déplacer le produit après l'achèvement de la borne minimale et avant l'achèvement de la borne supérieure, le chariot peut dans ce cas déplacer le produit vers la prochaine destination

-Cas3

Lorsque le chariot est disponible pour déplacer le produit après l'achèvement de la borne maximale, le produit est dans ce cas endommagé, et, l'ordonnancement n'est pas faisable.

Le temps d'attente du robot est δ_{ji}^k donné par :

$$\delta_{ji}^k = a_{j(i-1)}^l, \quad l < k$$

si l'opération encours et l'opération précédente appartiennent au même produit, et

$\delta_{ji}^k = a_{j(i-1)}^l$ - le temps dépensé des que le chariot a quitté la cuve l et jusqu'à son retour à la cuve l

pour le cas contraire.

Une valeur négative de δ_{ji}^k signifie que le chariot est venu déplacer le bain après l'écoulement du temps de séjour prévu pour l'opération considérée.

Pour garantir la qualité du produit, il est nécessaire de respecter la contrainte de fenêtre de temps pour chaque opération. Pour le cas où δ_{ji}^k est négatif, il faut s'assurer que $|\delta_{ji}^k| + a_{j(i-1)}^l \leq b_{j(i-1)}^l$, sinon le produit est défectueux et l'ordonnancement n'est pas faisable.

En effet l'expression $|\delta_{ji}^k| + a_{j(i-1)}^l$ présente la nouvelle notation du temps du traitement de l'opération $o_{j(i-1)}^l$ traitée dans la cuve T_j .

Comme déjà dit, l'ASC a été retenu comme une méthode de résolution pour la résolution de problèmes d'ordonnancement [Yih, 94], [Cheng et al, 97], [Hindi, 04]. Pour l'atelier de traitement de surfaces considéré, type SHMP Job Shop, un algorithme de satisfaction de contraintes fondé sur un nouveau critère qui est le calcul de temps d'attente et intégrant les temps de déplacement du chariot, est proposé afin de trouver l'ordonnancement des mouvements du chariot satisfaisant toutes les contraintes et cherchant un temps de cycle minimal.

II.4.3. L'algorithme de satisfaction de contraintes proposé

Au début de l'initialisation de l'algorithme, nous supposons que le tonneau (contenant un type de produit) est transporté de la cuve T_N vers la cuve T_k , le temps nécessaire pour effectuer ce déplacement est ainsi égal à la somme des déplacements élémentaires du chariot partant de T_k arrivant à T_N .

$$\theta_{kN} = \theta_{k,k+1} + \theta_{k+1,k+2} + \dots + \theta_{N-1,N}$$

Lorsque le tonneau est déposé à l'intérieur de la cuve T_k , la cuve quitte ce bain pour aller effectuer la tâche suivante.

Dès que tous les produits sont en ligne, si les tâches encours sont O_{ji} , $O_{j+1,i+2}$ et $O_{j+2,i+7}$, étant donnée que la contrainte d'antériorité au niveau des gammes doit être respectée, la tâche suivante sera sélectionnée aléatoirement parmi les trois possibilités suivantes, $O_{j,i+1}$, $O_{j+1,i+3}$ ou $O_{j+2,i+8}$. Le problème de la contrainte de capacité du robot apparaît lorsque deux produits sont prêts à quitter

leurs bails au même instant.

Considérant que le palan ne peut déplacer plus d'un tonneau à la fois, l'ordonnancement n'est donc possible que si l'un des deux produits est autorisé à rester dans sa cuve plus de temps. La cuve attribuée à la tâche suivante doit être vide parce que celle-ci ne peut pas recevoir plus qu'un tonneau; pour certains traitements, ce problème est toujours résolu car il peut y avoir plus d'un réservoir possible pour ce traitement, c'est le cas du traitement de rinçage. Ainsi, le principe de cette méthode consiste à trouver un séquençement des mouvements du chariot qui satisfait toutes les contraintes.

II.4.3.1. Description de l'algorithme

Entrée

Différentes opérations de chaque produit avec les fenêtres de temps.

Les temps de déplacements élémentaires du chariot

Niveau 1: Détermination de l'opération suivante dans l'ordonnancement

Définir, pour i allant 1 à M , le compteur des opérations (M étant le nombre des opérations)

Pas 1 Choisir les J premières opérations, chacune appartenant à un produit pour les itérations pour i allant 1 à J (J étant le nombre de produits)

Pas 2 Définir un vecteur avec J possibilités pour déterminer l'opération suivante ($1 \times J$)

Répéter le pas 2 jusqu'à trouver J différentes possibilités de ce vecteur

Pas 3 Choisir aléatoirement une tâche du vecteur de possibilités

Niveau 2: Tester les contraintes de capacité des ressources

Pas 4 Définir un vecteur d'occupation des cuves, $Indicateur = [I_1, I_2, \dots, I_N]$

Pas 5 Sélectionner la cuve attribuée T_k à la tâche sélectionnée

Pas 6 Vérifier la disponibilité du bain T_k

-Si $I_k=1$, la cuve k est occupée, le pas 3 est ensuite répété pour choisir une autre tâche de vecteur de possibilités

- Si toutes les cuves relatives, par rapport aux J tâches du vecteur de possibilités, sont occupées, l'algorithme est interrompu et un message est affiché «ordonnancement non faisable cause contrainte de ressources », sinon supprimer la

tâche sélectionnée de la liste des tâches et nous passons au niveau suivant.

Niveau 3: Calculer les temps de déplacement du chariot

Pas 7 Définir un tableau pour les temps des déplacements élémentaires du chariot

-Pour la tâche en cours O_{ji} et la tâche suivante $O_{j,i+1}$, appartenant au même produit, traitées respectivement dans les réservoirs T_k et T_{k+3} , le temps nécessaire à ce déplacement est :

$$\theta_{k,k+3} = \theta_{k,k+1} + \theta_{k+1,k+2} + \theta_{k+2,k+3}$$

Pas 8 -Pour la tâche en cours O_{ji} et la tâche suivante $O_{j+1,i+3}$, appartenant aux différents produits, traités respectivement dans les réservoirs T_k et T_{k+3} , la tâche précédente $O_{j+1,i+2}$ du produit $j+1$ est sélectionnée ainsi que le réservoir relatif T_{k-2} . Le palan se déplace à partir de T_k vers T_{k-2} pour récupérer le produit et le transporter vers T_{k+3} . Ainsi, le temps total de déplacement est donné par

$$\theta_{k,k+3} = \theta_{k-2,k-1} + \theta_{k-1,k} + \theta_{k,k+1} + \theta_{k+1,k+2} + \theta_{k+2,k+3}$$

Niveau 4 : Calculer le temps d'attente du chariot

Pas 9 - Pour la tâche en cours O_{ji} et la tâche précédente $O_{j,i-1}$ traitées respectivement dans les réservoirs T_k et T_{k-2} le temps d'attente nécessaire pour effectuer ce déplacement est

$$\delta_{ji}^k = a_{j,i-1}^{k-2}$$

- Pour la tâche encours O_{ji} , puisque la tâche précédente est $O_{j+1,i}$, la tâche précédente du produit j est sélectionnée $O_{j,i-1}$. Ces tâches sont traitées respectivement dans T_c , T_p et T_l . Le temps d'attente est :

$\delta_{ji}^c = a_{j,i-1}^l$ - Temps nécessaire pour effectuer les déplacements des que le chariot a quitté T_l jusqu'au son retour à T_l

-si $\delta_{ji}^c < 0$ pour $k = 1, \dots, N$, $i = 1, \dots, M_j$, $j = 1, \dots, J$, la flexibilité de fenêtre de temps est introduite et il est recommandé de respecter que $a_{j,i-1}^l + |\delta_{ji}^c| \leq b_{j,i-1}^l$, sinon dépassement de la borne maximale de temps de traitement du produit qui se trouve ainsi dégradé.

L'ordonnancement n'est donc pas faisable et par conséquent le temps de cycle est égal à 0.

-Pour le cas où $a_{j,i-1}^l + |\delta_{ji}^c| \leq b_{j,i-1}^l$, δ_{ji}^c devient égal à 0 et le nouveau temps de traitement α_{ji}^l de cette tâche est formulé comme suit : $\alpha_{ji}^l = a_{ji}^l + |\delta_{ji}^c|$

Niveau 5: Calculer la charge des bains

Pas 10 -Trouver le nombre d'occupation NO_k de chaque bain T_k
 -Multiplier NO_k par le temps de traitement attribué à T_k , ainsi la charge de la cuve est formulée comme suit : $Z_{ji}^k = NO_k * \alpha_{ji}^k$

Niveau 6: Sauvegarder les résultats

Pas 11 -Pour chaque itération $i = 1, \dots, M$,
 -Enregistrer le séquençement des opérations de l'ordonnancement S
 -Sauvegarder les cuves attribuées aux opérations de l'ordonnancement S
 -Sauvegarder les temps de traitement des opérations de l'ordonnancement S
 -Sauvegarder la charge des bains associées aux tâches de l'ordonnancement S
 -Sauvegarder les temps de déplacement
 -Sauvegarder les temps d'attente du chariot

Pas 12 -Calculer le maximum des charges des bains $Z = \max Z_{ji}^k$ pour $k = 1, \dots, N$
 -Calculer le temps total des déplacements du chariot $\sum_{j=1}^J \sum_{i=1}^{M_j} \theta_{ji}$ pour $i = 1, \dots, M_j, j = 1, \dots, J$
 -Calculer le temps total d'attente du chariot $\sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$ pour $k = 1, \dots, N, i = 1, \dots, M_j, j = 1, \dots, J$

-Calculer la charge du chariot $Y = \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k + \sum_{j=1}^J \sum_{i=1}^{M_j} \theta_{ji}$

-Calculer le temps du cycle $CT = \max(Y, Z)$

Pas 13 -Exécuter l'algorithme R fois pour trouver le temps de cycle minimum CT_{\min}

-
- Pour chaque répétition $r, r = 1, \dots, R$
 - Sauvegarder la variation des charges des baigns pour chaque ordonnancement $S^r, r = 1, \dots, R$
 - Sauvegarder la variation du temps d'attente du robot pour chaque ordonnancement $S^r, r = 1, \dots, R$
 - Sauvegarder la variation de la charge du robot pour chaque ordonnancement $S^r, r = 1, \dots, R$
 - Sauvegarder la variation du temps de cycle pour chaque ordonnancement $S^r, r = 1, \dots, R$

II.4.3.2. Résultats obtenus

Rappelons que l'atelier de traitement de surfaces considéré dans ce mémoire est dédié pour le traitement de 4 produits de la gamme de zingage, 2 produits de la gamme de nickelage et un produit de la gamme d'argentage. Vu la complexité combinatoire du problème avec ce nombre de gammes simultanées d'une part et le but de généraliser l'application de cet algorithme d'autre part, on a choisi de travailler avec 3 gammes, une de zingage, une de nickelage et une dernière d'argentage. Les résultats expérimentaux, obtenus par l'algorithme de satisfaction de contraintes proposé pour ce cas de figure, générés par Matlab 7.0, pour 10 implémentations, sont présentés comme suit.

Comme expliqué dans le niveau 4 de l'algorithme proposé, un ordonnancement $S^r, r = 1, \dots, R$ n'est faisable que si tous les temps d'attente du robot, associés aux opérations de cet ordonnancement, sont positifs. La faisabilité de l'ordonnancement dépend essentiellement des valeurs des temps d'attente du robot.

La figure II-3, montre que les valeurs d'attente du robot des 37 tâches sont uniquement positives pour l'implémentation 6. Les valeurs numériques des temps d'attente du robot, relatifs à la figure II-4, sont données dans le tableau II-6.

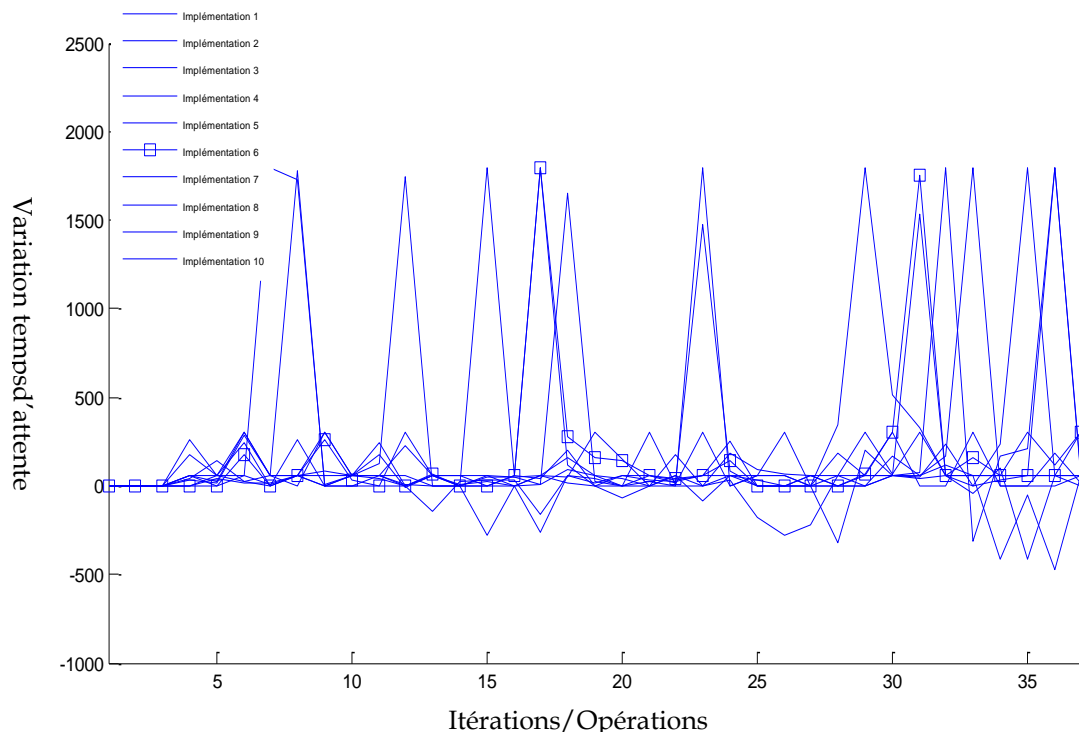


Figure II-4-Temps d'attente du robot par application de l'ASC

Tableau II-6-Les temps d'attente du chariot par application de l' ASC

Columns 1 through 15														
0	0	0	0.2608	0.0600	0.3000	0.0600	0	0.3000	0.0623	0.0397	0	0.0600	0.0543	0.0600
0	0	0	0.0426	0.1416	0.0227	0	0.2642	0	0	0	0.3000	0.0600	0	-0.2781
0	0	0	0	0.0600	0.2416	0	0.0600	0.0600	0.0600	0.1800	-0.0063	0.0600	0.0600	0
0	0	0	0.0600	0.0600	0.0251	0.0600	0.0600	0	0.0600	0.2468	0	-0.1423	0.0264	0.0315
0	0	0	0.0035	0.0441	0.0178	0.0185	0.0600	0.0836	0.0600	0.1241	1.7426	0	0	0.0600
0	0	0	0	0	0.1800	0	0.0600	0.2583	0.0302	0	0	0.0690	0	0
0	0	0	0.0600	0.0265	0.3000	0.0600	0.0563	0	0	0.0600	0	0.0600	0	0
0	0	0	0.0337	0.0173	0.2836	0.0600	1.7760	0	0.0600	0.0600	0.2276	0.0600	0.0109	0.0257
0	0	0	0.0600	0	0.0600	1.8000	1.7272	0.0073	0.0600	0.0600	0.0600	0	0	0.0498
0	0	0	0.1800	0.0600	0.0600	0.0059	0.0600	0.3000	0.0600	0.0600	0.0059	0	0	1.8000
Columns 16 through 30														
0.0472	0.0113	1.6543	0.0288	0	0	0.1800	0	0.0313	-0.1765	-0.2830	-0.2230	0.0600	0.0543	0.0600
0	-0.2583	0.0600	0.0151	0.0600	0.0301	0.0284	0.0600	0.2543	0	0	0.0600	0	0	0.0600
0	0.0104	0.0950	0.0600	0.0392	0	0.0600	0	0.1878	0.0950	0.0690	0.0600	0.0600	0	0.1647
0.0335	-0.1572	0.0562	0.0388	0	0.0600	0.0302	0.0600	0.0600	0.0355	0	0	0.0600	0.3000	0.0638
0	0.0600	0.1563	0.0600	0	0.0216	0	0	0.0600	0	0	0	-0.3231	0.2060	0.0600
0.0600	1.8000	0.2786	0.1624	0.1461	0.0600	0.0399	0.0600	0.1461	0	0	0	0	0.0690	0.3000
0.0600	1.8000	0.1155	0	-0.0690	0	0	1.8000	0	0.0600	0.3000	0	0.1858	0.0600	0.0600
0	0.0600	0.0173	0	0.0600	0.0600	0.0157	1.4730	0.0848	0	0	0.0583	0	0	0.0600
0.0498	0.0600	0.0464	0.3000	0.1488	0	0.0544	-0.0820	0.0600	0.0600	0.0544	0	0	0.0600	0.3000
0.0600	0.0407	0.2014	0	0	0.3000	0.0059	0.3000	0	0.0600	0.0600	0	0.3475	1.8000	0.5166
Columns 31 through 37														
0.0472	0.2360	0	0.2360	1.8000	0.0600	0.0600								
0.0729	1.8000	-0.3092	0.1656	0.2143	1.8000	0.0765								
0.0600	0.1647	1.8000	0	0	0.1843	0.0138								
0.3000	0.0600	0.0600	0.0600	0.0600	1.8000	0.0600								
1.5368	0.0600	0	0.0353	0.0600	0.0600	0								
1.7505	0.0600	0.1624	0.0600	0.0600	0.0600	0.3060								
0.0600	0.1140	0.0600	0.0600	0.3000	0.1140	0.3000								
0.0423	0.0600	-0.0398	0.0967	-0.4179	0.0600	0.0600								
0	0	0.3000	0	0.0034	0	0.0600								
0.3245	0.0600	0.0600	-0.4099	-0.0493	-0.4698	0.0600								

Ce tableau contient 37 colonnes présentant les valeurs numériques des temps d'attente au niveau des opérations et 10 lignes présentant le nombre d'implémentations de l'algorithme posé, d' où chaque ligne présente donc les valeurs des temps d'attente associés aux opérations d'un ordonnancement parmi les 10 obtenus.

Le tableau II-6 montre que les valeurs des temps d'attente du chariot sont négatives pour toutes les implémentations sauf l'implémentation 6, l'ordonnancement S^5 est ainsi le seul ordonnancement faisable.

En effet, la figure suivante présente le maximum des charges des baigns, la charge du chariot et le temps du cycle qui est égal au maximum de ces deux charges pour 10 implémentations. Le temps de cycle est alors l'intersection entre la courbe des maximums des charges des baigns et des charges du chariot.

Le résultat extrait de la figure II-4 est en effet confirmé dans la figure II-5 : le temps de cycle n'est différent de 0 que pour l'implémentation 6.

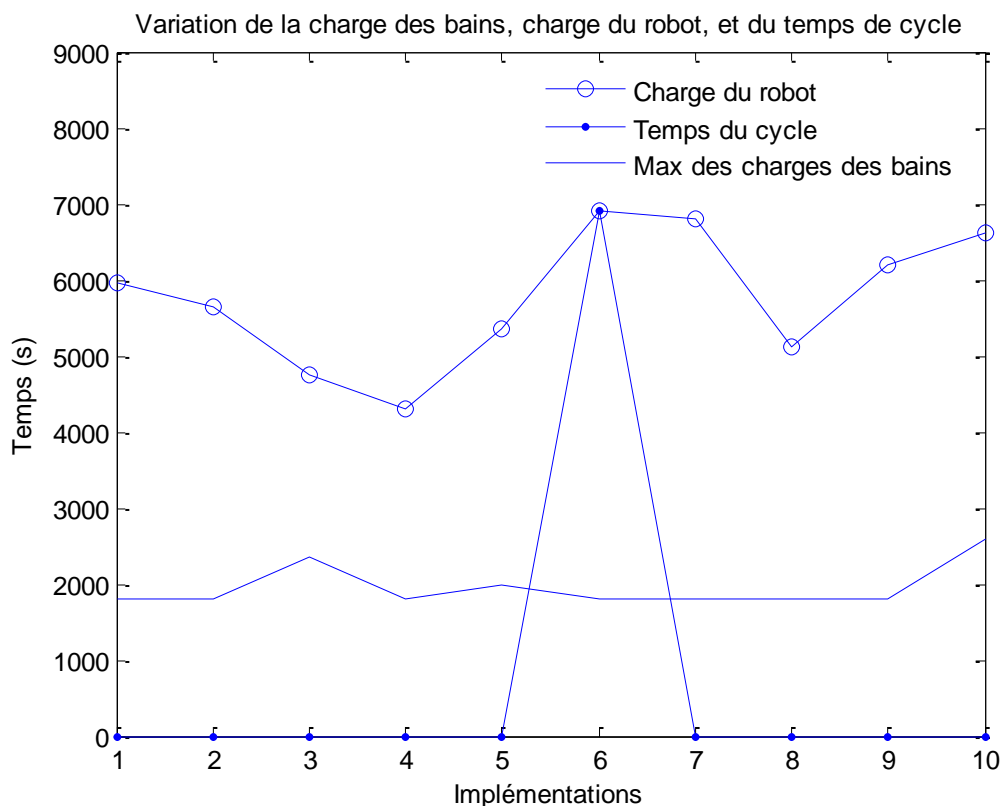


Figure II-5-Variations de la charge des baigns, de la charge du robot et du temps du cycle

Le tableau II-7 (respectivement II-8) contient 37 colonnes présentant les 37 opérations (respectivement 37 cuves) et 10 lignes présentant le nombre d'implémentations de l'algorithme posé, chaque ligne présente donc le séquençage des opérations (respectivement des cuves) associés à un ordonnancement parmi les 10 obtenus.

Tableau II-7-Séquencement des opérations par application de l' ASC

Columns 1 through 26																									
28	18	31	32	19	29	33	110	34	210	35	111	36	37	211	21	22	23	24	38	25	112	39	26	310	
28	18	31	29	210	32	33	19	211	21	22	110	111	23	34	112	35	113	114	36	37	24	38	25	39	
28	18	31	32	33	19	29	210	110	34	211	21	22	111	35	23	112	36	37	38	39	310	24	113	114	
28	18	31	29	19	210	32	33	34	35	211	21	36	37	38	22	110	23	39	111	112	113	24	310	114	
28	18	31	19	29	210	32	211	21	33	22	110	23	24	34	111	25	26	27	112	113	114	35	36	37	
28	18	31	19	32	110	29	33	111	112	34	210	35	36	37	211	21	113	22	23	24	25	114	115	38	
28	18	31	19	110	32	33	29	34	210	211	21	35	36	37	38	39	22	310	23	111	112	113	24	25	
28	18	31	19	29	210	110	211	32	21	22	111	23	24	25	26	33	27	112	113	34	35	114	115	36	
28	18	31	32	29	33	19	210	110	211	21	34	111	35	112	36	37	22	38	23	113	114	115	39	24	
28	18	31	29	19	210	211	32	21	33	34	110	35	111	22	23	24	112	113	36	37	25	38	39	114	
Columns 27 through 37																									
113	114	115	116	11	12	13	14	15	16	17															
115	26	116	11	27	12	13	14	15	16	17															
25	116	11	26	12	13	14	15	16	27	17															
26	27	115	116	11	12	13	14	15	16	17															
39	115	310	116	11	12	13	14	15	16	17															
11	26	12	13	27	39	14	15	16	17	310															
27	114	115	116	11	12	13	14	15	16	17															
116	11	38	39	310	12	13	14	15	16	17															
310	26	27	116	11	12	13	14	15	16	17															
27	115	116	11	310	12	13	14	15	16	17															

Tableau II-8-Séquencement des cuves par application de l' ASC

Columns 1 through 26																									
28	18	31	32	19	29	33	110	34	210	35	111	36	37	211	21	22	23	24	38	25	112	39	26	310	
28	18	31	29	210	32	33	19	211	21	22	110	111	23	34	112	35	113	114	36	37	24	38	25	39	
28	18	31	32	33	19	29	210	110	34	211	21	22	111	35	23	112	36	37	38	39	310	24	113	114	
28	18	31	29	19	210	32	33	34	35	211	21	36	37	38	22	110	23	39	111	112	113	24	310	114	
28	18	31	19	29	210	32	211	21	33	22	110	23	24	34	111	25	26	27	112	113	114	35	36	37	
28	18	31	19	32	110	29	33	111	112	34	210	35	36	37	211	21	113	22	23	24	25	114	115	38	
28	18	31	19	110	32	33	29	34	210	211	21	35	36	37	38	39	22	310	23	111	112	113	24	25	
28	18	31	19	29	210	110	211	32	21	22	111	23	24	25	26	33	27	112	113	34	35	114	115	36	
28	18	31	32	29	33	19	210	110	211	21	34	111	35	112	36	37	22	38	23	113	114	115	39	24	
28	18	31	29	19	210	211	32	21	33	34	110	35	111	22	23	24	112	113	36	37	25	38	39	114	
Columns 27 through 37																									
113	114	115	116	11	12	13	14	15	16	17															
115	26	116	11	27	12	13	14	15	16	17															
25	116	11	26	12	13	14	15	16	27	17															
26	27	115	116	11	12	13	14	15	16	17															
39	115	310	116	11	12	13	14	15	16	17															
11	26	12	13	27	39	14	15	16	17	310															
27	114	115	116	11	12	13	14	15	16	17															
116	11	38	39	310	12	13	14	15	16	17															
310	26	27	116	11	12	13	14	15	16	17															
27	115	116	11	310	12	13	14	15	16	17															

Le séquencement des opérations, les cuves associées, les temps de traitement, les temps d'attentes et les nouveaux temps de traitement, présentant la data de la solution obtenue, sont données dans le tableau II-9.

Tableau II-9-Data-Ordonnement obtenu par application de l' ASC

Opération	Bain	P_{ji}^k	δ_{ji}^k	α_{ji}^k
O_{13}	T_3	300	0	300
O_{23}	T_6	60	0	60
O_{34}	T_{21}	60	0	60
O_{35}	T_8	60	60	121
O_{24}	T_5	60	33.7	60
O_{25}	T_{17}	60	60	126
O_{14}	T_{12}	60	217	60
O_{15}	T_5	60	60	60
O_{16}	T_6	60	60	60
O_{17}	T_{12}	60	60	60
O_{18}	T_{16}	1800	60	1800
O_{19}	T_4	60	1800	60
O_{36}	T_{22}	60	-61	60
O_{26}	T_8	60	-66	60
O_{27}	T_9	60	60	152
O_{37}	T_{13}	180	33.8	180
O_{38}	T_{12}	60	180	205
$O_{1,10}$	T_{17}	60	6	280
O_{28}	T_{10}	1800	-92	1800
O_{39}	T_9	60	-145	60
$O_{3,10}$	T_{23}	60	60	60
$O_{1,11}$	T_{18}	60	-220	60
O_{29}	T_{11}	60	1754	94
$O_{1,12}$	T_{19}	60	44	60
O_{31}	T_1	300	18	300
$O_{1,13}$	T_{20}	60	13	60
$O_{2,10}$	T_7	60	-34	272
$O_{1,14}$	T_4	60	37	60
$O_{1,15}$	T_9	60	60	215

$O_{2,11}$	T_{23}	60	-212	301
O_{32}	T_{22}	60	211	60
O_{33}	T_5	60	60	60
O_{21}	T_1	300	-241	300
$O_{1,16}$	T_{23}	60	-155	153
O_{22}	T_{22}	60	211	60
O_{11}	T_1	300	-93	300
O_{12}	T_{21}	60	300	60

Pour mieux comprendre le résultat obtenu par l'algorithme proposé, les valeurs obtenues dans la ligne 5 et la ligne 14 du tableau II-9 peuvent être interprétées comme suit.

-Le produit est posé dans le bain T_8 pour effectuer l'opération O_{35} ayant un temps de traitement P_{ji}^k de 60 s.

-Quand le chariot est venu récupérer le produit du bain T_8 et le déplacer pour commencer la tâche suivante O_{36} , il a fait un retard de 61s par rapport au temps de traitement de la tâche O_{35} ; c'est à dire que le produit doit être immergé dans le bain T_8 pendant 121 sec au lieu de 60 s ; le temps d'attente du robot est ainsi égal -61s (ligne 14).

-A ce stade, on vérifie si le temps de traitement avec la valeur absolue du temps d'attente est inférieure à la marge maximale attribuée à la tâche O_{35} ; autrement dit cette marge doit être supérieure ou égale à 121 pour pouvoir poursuivre la construction de l'ordonnancement; une fois le test est positif, la nouvelle valeur de temps de traitement de l'opération O_{35} est égale à 121s (ligne 5). Pour le cas contraire, l'ordonnancement est interrompu (temps de cycle égal à 0) et on passe à l'implémentation d'après (refaire la construction de l'ordonnancement de nouveau).

Bien entendu l'approche proposée a réussi à donner au moins une solution admissible pour le problème SHMP étudié en intégrant les temps de déplacement mais après un nombre d'implémentations égal dans le cas présenté à 10, sachant qu'il est tout à fait possible de générer plusieurs solutions où on est obligé d'aller jusqu'à un nombre plus grand d'implémentations de pour trouver plus de solutions.

Une hybridation est présentée dans la section suivante dans ce sens pour l'amélioration des résultats trouvés.

II.4.4. L'algorithme de satisfaction de contraintes proposé avec hybridation

Un résultat est obtenu dans la partie précédente par application de l'algorithme de satisfaction de contraintes dans les problèmes d'ordonnancement, les temps de déplacements étant introduits. Pour être capable de résoudre les problèmes d'ordonnancement réels de manière plus efficace

[Jourdan et al, 09], l'hybridation peut être considérée. Les techniques hybrides, basées en général sur la combinaison des 2 méthodes, peuvent fournir de solutions de meilleure qualité. Dans cette partie, il est proposé l'incorporation d'heuristiques d'ordonnancement classiques dans l'algorithme de satisfaction de contraintes élaboré afin d'améliorer les résultats obtenus.

L'algorithme de satisfaction de contraintes, basé principalement sur le critère de calcul du temps d'attente du chariot, est hybridé avec des heuristiques classiques, déterminant le choix des opérations en fonction de leur temps de traitement remplaçant ainsi le choix aléatoire des opérations [Mhedhbi et al, 10].

Le choix des deux heuristiques d'hybridation proposées, pour résoudre le problème SHMP proposé, utilisant les données du problème, tableaux II-3, II-4, II-5, LPTF et SPTF, est basé sur le fait que la majorité des marges minimales des opérations des gammes étudiées sont des temps opératoires de courte durée. Avancer ces temps opératoires peut ainsi augmenter la fluidité du système alors qu'avancer ceux de longue durée peut le bloquer. Ce fait est constaté dans les résultats des études obtenus à la fin de ce chapitre.

L'ASC ainsi que ces deux heuristiques considérées sont introduits ci-après.
-L'algorithme de satisfaction de contraintes ASC, est basé sur la sélection arbitraire des opérations pour chaque itération.

-La deuxième approche est relative à l'hybridation de l'ASC avec l'heuristique LPTF (Longest Processing Time First) [Pinedo, 05], qui consiste en la sélection de la tâche ayant le plus long temps de traitement à chaque itération. Cette hybridation est notée ASC / LPTF.

Dans l'étape 1 de l'implémentation de l'ASC, la tâche ayant le plus long temps de traitement pour chaque produit est sélectionnée à chaque itération. Un vecteur avec ces possibilités est ensuite établi pour trouver la prochaine tâche. Ces tâches sont rangées dans un ordre décroissant en fonction du temps de traitement. Le premier élément du vecteur est choisi et si la cuve associée à cette tâche est occupée, le deuxième élément du vecteur est sélectionné jusqu'à avoir la tâche satisfaisant la contrainte de ressources à l'étape 6.

-La troisième approche est relative à une hybridation de l'ASC avec SPTF (Shortest Processing Time First) [Pinedo, 05], noté ASC / SPTF, similaire à celle précédente ASC / LPTF ; la tâche, ayant cette fois le temps de traitement le plus court, est sélectionnée pour chaque itération.

II.4.4.1. ASC versus ASC / LPTF

Pour l'hybridation ASC / LPTF, les valeurs des temps d'attente de levage des 37 tâches ne sont positifs que dans l'implémentation 5 comme le montre le tableau II-10. Il n'y a donc qu'un calendrier réalisable des opérations générées par ASC/LPTF dans l'implémentation 5, figure II-6.

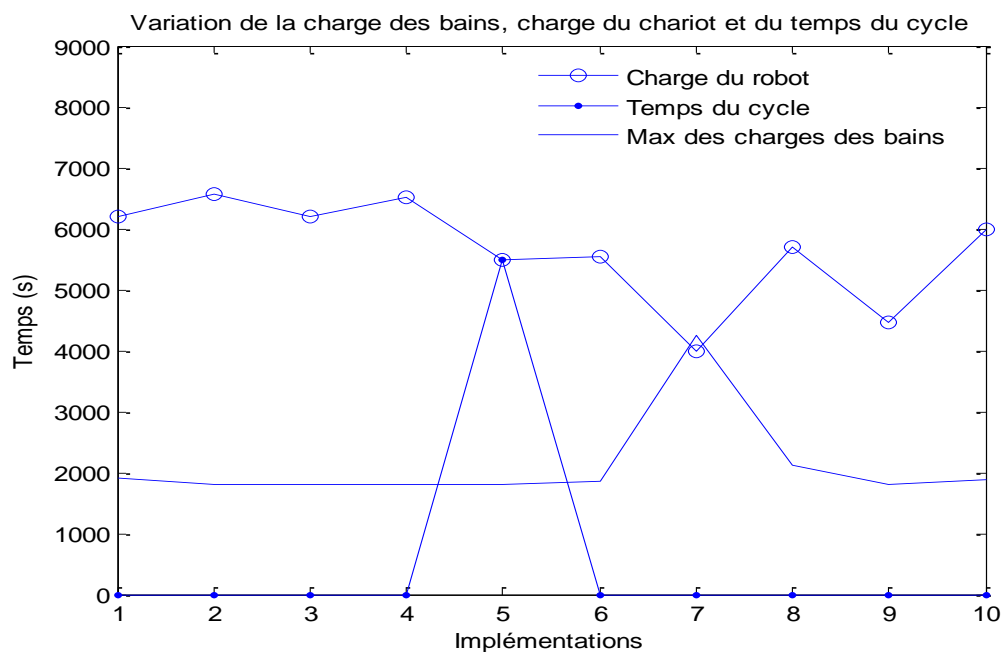


Figure II-6-Variations de la charge des baigns, de la charge du robot et du temps du cycle pour ASC/LPTF

Tableau II-10-Les temps d'attente du chariot par applicaion de l' ASC/LPTF

Columns 1 through 15														
0	0	0	0.0092	0.3000	0.0600	0.3000	0.0600	0.0600	0	0.0600	0.1800	0.0600	0	0
0	0	0	0.0600	0.1800	0	0.0600	0.0600	0	0	0.1384	0	0	0.0506	0
0	0	0	0.0338	0	1.7737	0	0	0.0600	0	0	0.3000	0.1121	0	0.3000
0	0	0	0.0600	0.1800	0.0212	0.0235	0.0600	0.0600	0.3000	0	0.0600	0.0600	0.0600	0.0600
0	0	0	0.0460	0.0513	0.0408	0.0600	0.3000	0	0.3000	0	0.0529	0.0476	0	0.0086
0	0	0	0.0408	0.0600	0	0	0.0600	1.8000	0.1909	0.0600	0	0	0.3000	0.0757
0	0	0	0.0600	0.2212	0	0.0600	1.6393	0.0232	0.0019	0.1800	0.0600	0.2679	0.0232	0
0	0	0	0.0600	0	0	0.0102	0.1800	1.7172	0	0	0	-0.1276	0.0600	0.0105
0	0	0	0.0321	0	0.0600	0	0.0123	0.0015	0.1938	0.0495	0.0600	1.8000	0	0
0	0	0	0.0600	0.0040	0.0600	0.0350	0	1.7190	0.0600	0	0.0600	0.0600	0	0.0221
Columns 16 through 30														
0	1.8000	0	0	0.0600	0	0	0	-0.2290	0.0600	0	1.5209	0.0600	0.1860	-0.4726
0.0891	0	0.0600	0	0.2723	0	-0.4139	0	0.1318	0.3020	-0.2538	0.3020	1.8000	0.0600	0.0600
0	0	0.0439	0.1822	0.0488	0.2123	-0.2822	0	0.0273	0.0600	0	0.0273	0	0	0.0273
1.8000	0.0106	0.0600	0	0.0348	0.0600	0.0750	-0.1271	0.0835	0	0	0	1.3933	0.1460	0.0600
0.0600	0	0	0.1800	0.0158	0	0	0.0600	0.0818	0	0.3000	0.1216	0.3000	0.0600	0.1093
0.0736	0	0.0515	0.0321	0.0406	-0.3879	-0.1746	0.0736	0.0600	0.0001	0.0600	0.0600	0.0392	0	-0.2994
0	-0.3019	0.0283	0	0.0283	0.0600	-0.3075	1.7617	0	0	-0.2576	0	0.0075	0	-0.6305
0.3000	0.0419	0.0600	0	0.0600	0	-0.2910	0.0600	0	-0.3841	1.7912	0.0600	0	0.1493	0.0600
0.0600	0	0	0.0600	0.0600	-0.0366	0.0406	-0.3006	0.0600	0	0	0.0652	0	0.0600	1.3268
1.8000	0.0600	0.1659	0	0.0015	0	-0.0803	0.0600	0	0.0600	0.0156	0	0	0.0600	0.0600
Columns 31 through 37														
0.0957	0.0600	0	0.1218	0.4163	0.4795	0.0898								
0.0600	1.6648	0.0423	0.0442	0.3020	0.2501	0.3862								
0.2804	1.8000	0.1228	0.0600	0.0600	0.0600	0.0600								
0.0600	0.0600	0.0600	0.1219	0.0600	0.4459	0.3000								
0.0600	0.3743	1.8000	0.0600	0.1216	0.0600	0.0600								
0.0600	1.8000	0.0830	0	0.0600	0.3000	0.0757								
-0.7917	0	0.0600	-0.2979	0.0600	0.8517	0.0600								
0	0.1974	0	0	0.3000	0.4242	0.0600								
-0.4826	0	0	0.0600	0.3086	0.0600	0								
0	0	0.2649	0	0.4427	0	0.0600								

Cette hybridation n'améliore pas les résultats obtenus par l'application d'ASC. En effet, les tableaux II-3, II-4, II-5 montrent que les plus longues tâches respectivement pour le cycle de zingage, le cycle de nickelage et celui d'argentage sont O_{18} , O_{28} et O_{38} .

Par conséquent, même si on augmente le nombre d'implémentations de l'algorithme afin de chercher plus de combinaisons pour atteindre le meilleur des résultats, les tâches générées pour les 3 premières itérations, comme le montre le tableau II-11, sont toujours les mêmes quelque soit le nombre d'implémentations, . Il n'y a donc pas suffisamment de combinaisons possibles lors de la construction de l'ordonnancement. La ligne 5 du tableau II-11 et du tableau II-12 présente le séquençement des opérations et celui des bains associés pour la solution faisable obtenue par application du ASC/LPTF.

Tableau II-11-Séquence des opérations par application de l' ASC/LPTF

28	18	31	32	19	29	210	211	21	33	34	22	23	35	110	36	37	111	112	113	38
28	18	31	32	19	110	111	112	113	33	29	114	210	211	21	115	22	23	116	11	12
28	18	31	32	33	29	34	210	19	35	110	211	21	111	112	22	113	36	37	38	114
28	18	31	29	210	19	110	32	111	112	211	21	22	113	23	24	25	26	114	115	27
28	18	31	32	29	33	19	110	111	210	211	21	22	112	113	114	115	34	23	116	11
28	18	31	32	29	19	210	110	33	111	112	34	113	114	115	211	21	22	35	36	37
28	18	31	29	32	19	210	33	34	211	21	110	22	111	35	112	23	113	36	37	114
28	18	31	32	33	29	210	34	211	21	22	19	35	23	110	36	37	111	24	112	25
28	18	31	32	33	19	29	210	34	211	21	22	23	24	25	26	27	35	36	37	38
28	18	31	19	32	29	33	34	110	111	210	35	36	37	112	113	38	39	211	21	310
24	39	310	114	25	26	27	115	116	11	12	13	14	15	16	17					
13	14	15	16	17	34	24	25	26	27	35	36	37	38	39	310					
23	24	39	310	115	116	11	25	26	27	12	13	14	15	16	17					
33	116	11	12	13	34	14	15	16	17	35	36	37	38	39	310					
35	24	12	13	36	37	14	15	16	25	38	26	39	310	17	27					
23	38	39	24	310	25	116	11	12	13	14	15	16	17	26	27					
115	24	25	38	116	11	26	27	12	13	14	15	16	17	39	310					
26	38	113	114	27	115	116	11	39	12	13	310	14	15	16	17					
110	111	112	113	39	310	114	115	116	11	12	13	14	15	16	17					
22	114	23	24	115	116	11	12	13	14	15	16	17	25	26	27					

Tableau II-12-Séquence des cuves par application de l' ASC/LPTF

10	16	1	15	9	8	4	23	1	5	17	4	5	8	2	22	13	18	19	20	21
10	14	1	9	2	21	18	19	20	6	8	4	7	23	1	15	17	5	23	1	4
10	16	1	4	5	8	21	15	12	8	4	23	1	18	19	4	20	22	13	2	15
10	14	1	8	9	12	21	2	18	19	23	1	15	20	6	5	12	8	4	12	21
10	14	1	22	8	5	2	22	18	12	23	1	21	19	20	9	7	22	6	23	1
10	14	1	22	8	9	17	21	5	18	19	4	20	2	7	23	1	12	8	2	13
10	14	1	8	4	17	22	5	4	23	1	2	7	18	8	19	6	20	4	13	15
10	16	1	7	6	8	12	9	23	1	12	7	8	6	4	17	13	18	5	19	4
10	16	1	12	5	7	8	21	15	23	1	2	5	6	2	8	2	8	21	13	21
10	14	1	22	7	8	5	9	12	18	22	8	7	13	19	20	2	15	23	1	23
6	15	23	12	17	8	21	7	23	1	15	3	17	5	6	7					
3	12	5	6	9	4	5	21	8	21	8	2	13	12	15	23					
5	6	9	23	22	23	1	17	8	4	21	3	2	6	5	22					
6	23	1	4	3	2	12	6	5	22	8	15	13	7	4	23					
8	5	4	3	7	13	15	6	5	21	15	8	12	23	4	21					
6	9	17	5	23	4	23	1	15	3	21	5	6	9	8	17					
2	5	22	15	23	1	8	12	17	3	9	5	6	9	4	23					
8	22	20	15	21	7	23	1	2	9	3	23	17	5	6	4					
9	18	19	20	4	23	17	12	23	1	22	3	22	6	5	4					
21	15	6	5	22	23	1	2	3	7	6	5	12	17	8	22					

II.4.4.2. ASC versus ASC / SPTF

Cette heuristique engendre, contrairement à l'ASC, plusieurs ordonnancements faisables comme le montre la figure II-7.

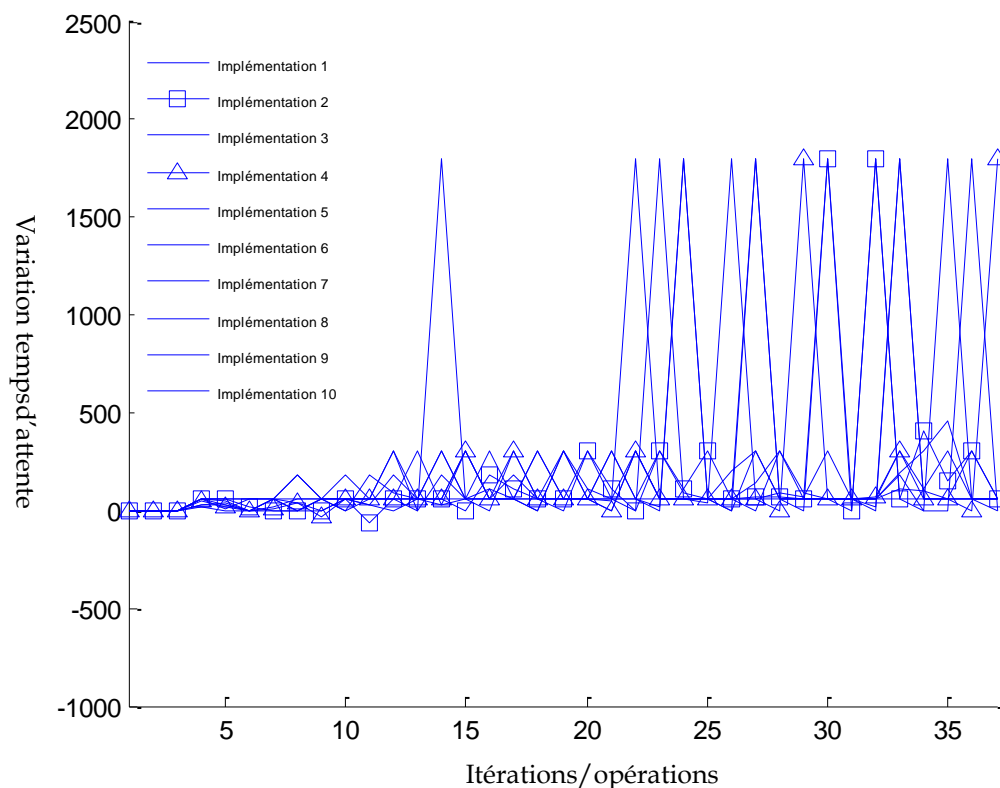


Figure II-7- Les temps d'attente du robot par application de l'ASC/SPTF

Il n'y a des valeurs négatives de temps d'attente du chariot qu'au niveau de l'itération 2 et 4, pour le reste des itérations les valeurs de temps d'attente du robot sont toutes positives. L'ordonnancement non faisable est ainsi indiqué.

Les valeurs numériques des temps d'attente relatifs aux implémentations générées par ASC/SPTF sont données dans le tableau II-13 suivant.

Tableau II-13-Les temps d'attente par application de l' ASC/SPTF

Columns 1 through 15														
0	0	0	0.0248	0.0300	0	0	0	0.0600	0.1800	0.0600	0.0600	0.0600	0.3000	0.0600
0	0	0	0.0600	0.0600	0.0177	0	0	0	0.0616	-0.0611	0.0600	0.0600	0.0600	0
0	0	0	0.0252	0.0353	0	0.0161	0.0600	0.0600	0.0600	0.0270	0.1800	0.0600	0.0600	0.0600
0	0	0	0.0460	0.0215	0	0.0128	0.0338	0.0363	0.0600	0.0600	0.0600	0.0600	0.0600	0.3000
0	0	0	0.0600	0.0269	0	0.0600	0.1800	0.0600	0.0600	0.0600	0.3000	0.0600	0.3000	0.0600
0	0	0	0.0198	0.0000	0	0.0600	0	0.0600	0	0.1800	0.0848	0.0600	0.0600	0.3000
0	0	0	0.0600	0.0600	0.0600	0.0600	0.1800	0.0600	0.0600	0.0600	0.3000	0	1.8000	0.0600
0	0	0	0.0600	0.0600	0.0600	0.0600	0.0355	0	0.0600	0.0302	0	0.0600	0.0285	0.0600
0	0	0	0.0600	0.0498	0	0.0600	0	0.0600	0.0600	0.0600	0.0600	0.3000	0.0265	0.3000
0	0	0	0.0600	0.0110	0.0600	0.0600	0.0600	0.0600	0	0.0600	0.0600	0	0.1800	0.0600
Columns 16 through 30														
0	0.3000	0.0600	0.3000	0.1078	0.0600	0.0600	0.0157	1.8000	0.0600	0.0600	0.1421	0.3000	0.0600	0.0600
0.1800	0.1111	0.0600	0.0600	0.3000	0.1111	0	0.3000	0.1118	0.3000	0.0600	0.0719	0.0719	0.0600	1.8000
0.3000	0.0600	0.0600	0	0.3000	0.0600	0.3000	0.0600	0.0600	0.0600	0.0600	1.8000	0.0600	0	1.8000
0.0600	0.3000	0.0600	0.0600	0.0600	0	0.3000	0.0600	0.0600	0.0600	0.0600	0.0600	0	1.8000	0.0600
0.0600	0.0600	0	0.3000	0.0600	0.0600	0	1.8000	0.0600	0.0600	0.2012	0.3000	0.0600	0.0600	0.0600
0.0600	0.0600	0	0.3000	0.0600	0.3000	0.0600	0.0600	0.0600	0.0388	1.8000	0.0600	0.0848	0.0643	0.3000
0.0600	0.0600	0.3000	0.0600	0.0600	0.0600	0.0600	0	1.8000	0.0600	0.0600	0.0600	0.0600	0.0600	0.0600
0.0600	0.1800	0.0600	0.0600	0.0600	0.3000	0.0600	0.3000	0.0876	0.0600	0.0600	0	0.3000	0.0876	0.0600
0.0600	0.0600	0.0600	0.0600	0.0600	0	0.3000	0.0613	0.0600	0.0600	0	0.3000	0.0600	0.0600	0.0600
0.1121	0.0600	0.3000	0.0600	0.0600	0.0473	1.8000	0.0600	0.0600	0.0600	0	1.8000	0.0600	0.0600	0.0600
Columns 31 through 37														
0.0600	0	1.8000	0.0600	0.0600	0.0600	0.0600								
0	1.8000	0.0600	0.4102	0.1477	0.3000	0.0600								
0.0600	0.0600	0.1943	0.3000	0.4537	0.0600	0.0600								
0.0600	0.0600	0.3000	0.0600	0.0600	0	1.8000								
0	1.8000	0.0600	0.0600	0.0600	0.0600	0.0600								
0.0600	0.0170	1.8000	0.0600	0.0600	0.0600	0.0600								
0.0600	0.0600	0.3000	0.1054	0.3000	0.0600	0.0600								
0.0600	0.0600	0.0600	0	0	1.8000	0								
0.0600	0.0600	0.1800	0	1.8000	0.0613	0								
0.0600	0.0600	0.1121	0.1041	0.0600	0.3000	0.0600								

Les valeurs du temps de cycle obtenues par ASC / SPTF sont présentées dans la figure II-8; il n'y a que deux valeurs nulles pour le temps du cycle relatives aux itérations 2 et 4 et le reste des solutions sont réalisables, confirmant le résultat extrait de la figure II-7 et du tableau II-13 précédents.

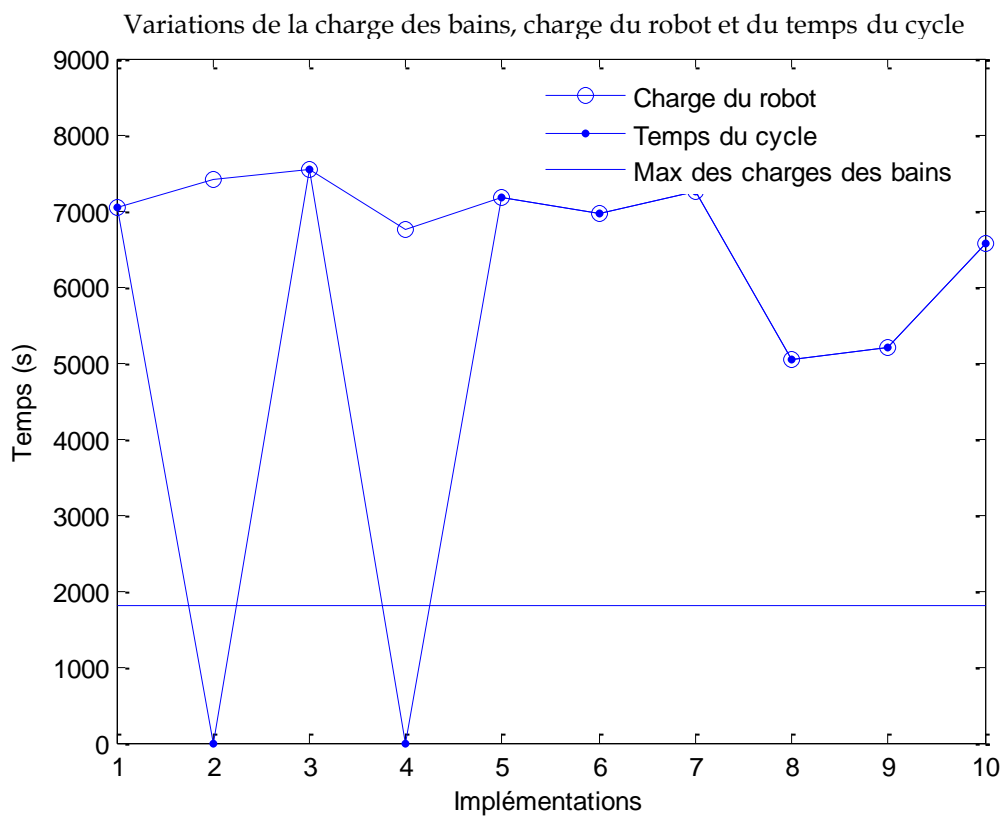


Figure II-8-Variations de la charge des baigns, charge du robot et du temps du cycle pour ASC/SPTF

Le séquencement des opérations ainsi que les cuves relatives aux ordonnancements générés par ASC/SPTF sont donnés dans les tableaux II-14 et II-15.

Tableau II-14-Séquence des opérations par application de l' ASC/SPTF

Columns 1 through 26																								
27	113	310	114	115	116	31	32	33	34	35	36	37	38	39	11	12	13	14	15	16	17	28	29	210
23	34	111	24	25	26	27	112	35	36	113	114	115	116	37	38	39	310	31	32	33	11	12	13	14
26	38	112	39	27	113	310	114	115	116	11	12	13	14	15	16	17	31	32	33	34	35	36	37	18
34	111	210	35	36	112	211	113	114	115	116	37	38	39	310	11	12	21	22	23	24	25	26	27	13
22	36	110	23	111	24	112	25	26	27	113	114	115	116	37	38	39	310	11	12	13	14	15	16	17
12	23	310	24	25	26	27	13	14	15	16	17	31	32	33	34	35	36	37	38	39	18	19	110	111
26	110	310	111	112	27	113	114	115	116	11	12	31	32	33	34	35	36	37	38	39	13	14	15	16
27	39	115	116	310	11	12	31	32	33	34	35	36	37	38	13	14	15	16	17	18	19	110	111	112
25	111	310	26	112	27	113	114	115	116	31	32	33	34	35	36	37	38	39	11	12	13	14	15	16
12	25	310	26	27	13	14	15	16	17	31	32	33	34	35	36	37	38	39	18	19	110	111	112	113
Columns 27 through 37																								
21	22	23	24	25	26	18	19	110	111	112														
16	17	18	19	110	28	29	210	211	21	22	23	24	25											
110	111	28	29	210	211	21	22	23	24	25														
15	16	17	31	32	33	28	29	18	19	110														
32	33	34	35	28	29	210	211	21	18	19														
113	114	115	116	11	28	29	210	211	21	22														
28	29	210	211	21	22	23	24	25	18	19														
114	28	29	210	211	21	22	23	24	25	26														
28	29	210	211	21	22	23	24	18	19	110														
115	116	11	28	29	210	211	21	22	23	24														

Tableau II-15-Séquence des cuves par application de l' ASC/SPTF

Columns 1 through 26																								
17	20	23	15	12	23	1	9	5	7	8	9	13	2	15	1	9	3	12	6	5	9	10	11	17
6	12	18	5	9	8	2	19	8	9	20	17	22	23	13	15	7	23	1	15	5	1	12	3	9
8	4	19	2	12	20	23	22	9	23	1	2	3	9	5	6	15	1	7	5	17	8	17	13	14
21	18	2	8	9	19	23	20	4	22	23	13	7	17	23	1	17	1	21	6	5	9	8	4	3
15	22	2	5	18	6	19	17	8	4	20	12	2	23	13	9	15	23	1	17	3	17	5	6	12
9	5	23	6	7	8	21	3	15	5	6	12	1	15	5	4	8	2	13	15	7	16	2	22	18
8	9	23	18	19	22	20	2	12	23	1	15	1	21	6	2	8	21	13	21	7	3	4	6	5
2	9	21	23	23	1	7	1	9	6	9	8	9	13	22	3	21	5	6	17	16	9	21	18	19
4	18	23	8	19	22	20	2	4	23	1	17	5	12	8	17	13	15	4	1	7	3	21	5	6
2	21	23	8	21	3	12	6	5	2	1	17	6	22	8	22	13	22	15	14	4	22	18	19	20
Columns 27 through 37																								
1	22	6	5	4	8	14	7	12	18	19														
5	17	14	7	4	10	11	12	23	1	17														
15	18	10	11	15	23	1	2	6	5	17														
5	6	22	1	21	6	10	11	14	15	4														
15	6	15	8	10	11	15	23	1	14	4														
20	4	22	23	1	10	11	12	23	1	4														
10	11	4	23	1	17	5	6	17	16	15														
7	10	11	4	23	1	9	5	6	12	8														
10	11	9	23	1	22	5	6	14	9	17														
2	23	1	10	11	7	23	1	21	5	6														

La comparaison des résultats obtenus par les 3 approches ont été présentées dans la première partie de ce chapitre est envisagée dans ce paragraphe. Un algorithme de satisfaction de contraintes, basé sur le calcul des temps d'attente et introduisant les temps de déplacement a donné naissance à une solution admissible sur un ensemble de 10. Une hybridation avec deux heuristiques, SPTF et LPTF, imposant ainsi un critère de sélection de l'opération à ordonnancer est proposée pour améliorer ce résultat. L'hybridation avec SPTF a donné de meilleurs résultats en donnant une moyenne de 8 solutions admissibles sur un ensemble de 10.

L'optimisation par l'exploitation d'une métaheuristique est introduite dans la section suivante. Notre choix a été porté sur les algorithmes génétiques, bien connue compte tenu par son efficacité pour l'obtention des optima globaux et pour son exploitation ultérieure dans divers travaux sur les ateliers de traitement de surfaces, [Lim, 97], [Pezzella et al, 08].

II.5. Optimisation utilisant les Algorithmes Génétiques (AGs)

Depuis l'idée originale proposée par Holland, [Holland, 75], le développement des algorithmes génétiques, dans les trois dernières décennies a connu un essor considérable. Les AGs ont été également adoptées avec succès pour résoudre le problème HSP, comme en témoigne le nombre croissant de papiers sur le sujet [David, 91], [Goldberg, 94], [Lim, 97], [Ross, 97], [Chen et al, 99], [Knosala et al, 01], [Hindi et al, 04], [Watanabe et al , 05], [Pezzella et al,08]. La majorité de ces travaux trouve des difficultés au niveau de la ressource du transport et considère les temps des déplacements du chariot négligeables ou bien constants ce qui relativise l'importance des résultats trouvés. L'apport de l'algorithme proposé, dans ce mémoire, même pouvant être considéré classique, est l'intégration des temps de déplacement dans la construction de l'ordonnancement afin de générer une solution la plus proche possible de la réalité.

II.5.1. Terminologie

Les algorithmes génétiques emploient un vocabulaire emprunté à la génétique naturelle. En effet, la communauté scientifique, utilisant les algorithmes génétiques utilise un certain nombre d'analogies issues du monde biologique pour illustrer des idées, [Kaabi, 04].

Définition 1. (Gène) Un gène est une suite de bases azotées qui contient le code d'une protéine donnée. Chaque gène est la suite de plusieurs symboles, appelés *allèles*, codant la valeur d'une variable.

Dans le cas général, un gène correspond à un seul symbole (0 ou 1 dans le cas binaire). Une mutation changera donc systématiquement l'expression du gène muté.

Définition 2. (Chromosome) Un chromosome est constitué d'une séquence finie de gènes qui

peuvent prendre des valeurs, appelées allèles, qui sont prises dans un alphabet qui doit être judicieusement choisi pour convenir au problème étudié.

Définition 3. (Individu) On appelle individu une des solutions potentielles. Dans la plupart des cas un individu est représenté par un seul chromosome, appelé indifféremment individu et chromosome.

Définition 4. (Population) On appelle population l'ensemble des solutions potentielles qu'utilise les AGs.

Définition 5. (Génération) On appelle génération l'ensemble des opérations permettant de passer d'une population Π_i à une population Π_j . Ces opérations sont généralement : sélection des individus de la population courante, l'application des opérateurs génétiques, et l'évaluation des individus de la nouvelle population. Par abus de langage, on peut aussi appeler $i^{\text{ème}}$ génération l'ensemble des individus après i itérations de l'algorithme.

Le tableau II-16 présente une analogie entre un individu dans une population et une solution d'un problème dans un ensemble de solutions.

Tableau II-16-Analogies entre la génétique naturelle et les problèmes d'ordonnancement

<i>Génétique Naturelle</i>	<i>Problèmes d'ordonnancement</i>
Population	Ensemble de solutions : ensemble des ordonnancements faisables
Individu (chromosome = séquence finie de gènes) parmi la population	Une solution : un ordonnancement parmi les ordonnancements faisables
Gene (une suite d'allèles)	Une tâche de l'ordonnancement (l'ensemble de paramètres codant la tâche)
Allèle	Un paramètre attribué à la tâche (ex : temps de traitement, position de la tâche dans le séquencement des opérations)

II.5.2. Fonctionnement des AGs

La mise en œuvre des AGs commence par calculer une population initiale, c'est à dire la première génération. La création de la population initiale peut être aléatoire ou en appliquant une heuristique. Nous supposons que la population initiale contient des individus noté également chromosome.

À chaque génération, un nouvel ensemble d'individus est créé en utilisant les meilleures parties des éléments de la génération du précédent. L'algorithme s'arrête si un nombre prédéterminé de générations ou un délai déterminé est atteint, ou si une valeur optimale calculée à l'avance d'un individu est atteinte à l'avance.

Les algorithmes génétiques ne sont pas purement aléatoires. Ils exploitent efficacement les informations précédemment obtenues pour juger la position des nouveaux points à explorer, avec l'espoir d'améliorer la performance.

Les différents opérateurs utilisés dans les algorithmes génétiques sont la sélection, le croisement et la mutation.

II.5.2.1. Le codage dans les ateliers de type Jop Shop

Dans la littérature, [Tangour, 07], plusieurs codages ont été proposés pour le job shop classique et le job shop flexible. Yamada a proposé un codage qui représente les dates de fin des opérations pour chaque tâche [Yam, 03]. Tamaki a utilisé un codage binaire traduisant la représentation de la solution en graphe disjonctif [Tam, 92]. Dans le codage de Kobayashi, les séquences des opérations par machine sont uniquement représentées [Kob, 95]. Enfin, Portmann propose un codage indirect sous forme d'écriture matricielle de la présence d'un enchaînement entre deux opérations consécutives [Por, 88].

Mesghouni propose aussi deux codages représentant des extensions de codages existants [Mes, 99]. Le premier utilise le codage de Kobayashi en ajoutant pour chaque opération d'une séquence, l'ordre de l'opération dans la gamme et sa date de début d'exécution. Le deuxième, une extension du codage de Yamada, introduit la machine à laquelle sont affectées les opérations pour chaque tâche. Enfin Kacem [Kac, 03], propose trois codages, un codage opérations/machines qui donne les dates de début et de fin de chaque opération sur la machine à laquelle est

affectée l'opération, un deuxième codage liste des opérations qui représente l'ordonnancement dans un tableau de trois colonnes: opération, machine capable d'exécuter l'opération et une troisième colonne pour les dates de fin.

Le troisième codage est celui des séquencements des jobs représentant l'ordonnancement en n colonnes (où n est le nombre maximum d'opérations que peut contenir un job). Chaque colonne représente les tâches à ordonnancer sous forme d'une liste de x cellules, (x étant le nombre de tâches. Chaque cellule est codée par le numéro de la tâche, la machine à laquelle l'opération est affectée, la date de début et de fin de l'opération.

Pour le job shop flexible, Ghedjati propose un codage basé sur l'affectation d'une heuristique de choix à chaque ressource [Ghe, 94]. Ces heuristiques permettent à la ressource de choisir la tâche à réaliser. Cette représentation un chromosome contient aussi autant de gènes que de ressources.

Pour le job shop flexible ayant plusieurs contraintes, une représentation augmentée des connaissances spécifiques au problème est proposée par [Bruns, 93], elle consiste à inclure, dans la conception même du chromosome, toutes les informations particulières concernant le problème à traiter. Ce codage est très pratique pour les problèmes ayant de nombreuses contraintes. Nous pouvons tous les inclure, tout au moins une partie d'entre eux, dans la conception du chromosome.

II.5.2.2. Opérateur de sélection

L'objectif de l'opérateur de sélection est de choisir des individus qui peuvent survivre et / ou de reproduire eux-mêmes pour transmettre leurs caractéristiques à la génération suivante. L'opérateur de sélection est fondé sur le principe de conservation des individus les plus adaptés et l'élimination de ceux moins. Aucun opérateur de sélection n'est absolument parfait, un risque de favoriser un certain nombre d'individus existe toujours et pourrait être un inconvénient réel.

II.5.2.3. Opérateur de croisement

L'opérateur de croisement assure la recombinaison des gènes des parents pour former les descendants.

Il a pour but d'enrichir la diversité de la population en intervenant au niveau de la structure des

chromosomes. Généralement les croisements sont envisagés avec deux parents et génèrent deux enfants. Dans ce paragraphe, nous allons décrire successivement les principaux opérateurs de croisement utilisés dans l'ordonnancement [Mes, 99].

- Croisement PMX

Le PMX (Partially Mapped Crossover) croisement à deux points, a été suggéré pour résoudre le problème de voyageur et des problèmes de type job shop. Une partie de la chaîne d'un parent est plaquée sur une portion de la chaîne d'un autre parent et les restes des informations sont échangés.

- Croisement MPX

Ce croisement (Maximal Preservative Crossover) a été proposé pour le problème du voyageur de commerce. Cet opérateur consiste à insérer un segment du chromosome parent 1 dans le chromosome parent 2 afin que le chromosome résultant soit le plus proche possible de ses parents.

- Croisement d'ordre OX

Pour le croisement OX (Order Crossover), deux points de croisement sont choisis au hasard; la zone située entre ces positions est copiée dans les mêmes positions de la descendance. Les symboles copiés sont supprimés de l'autre parent, et les autres symboles sont hérités, en commençant par la première position suivant le second point de croisement. Après avoir changé les rôles des parents, la même procédure est appliquée pour produire la deuxième. On peut distinguer le LOX (Linear Order Crossover) et le UOX (Uniform Order Crossover).

- Croisement d'ordre LOX

Le LOX (Linear Order Crossover) est une version modifiée de l'opérateur OX, et le chromosome est considéré linéaire à la place de la circulaire. Deux points de croisement sont d'abord choisis au hasard. Les symboles du premier parent entre les deux points de croisement sont supprimés dans le deuxième parent. Les trous créés sont glissés du centre vers les extrémités ; la zone de

croisement du parent 2 est alors vidée et prend alors les symboles de la zone du croisement du parent 1, le premier enfant est ainsi produit. Après avoir changé les rôles des parents, la même procédure est appliquée pour produire le deuxième enfant.

- Croisement d'ordre UOX

Le UOX (Uniform Order Crossover) consiste à sélectionner de façon aléatoire un ensemble de positions. L'ordre, dans lequel les éléments des positions sélectionnées apparaissent dans un parent est imposé à l'autre parent pour produire les deux enfants.

- Croisement basé sur les positions UPX

Le UPX (Uniform Position Crossover), consiste à choisir aléatoirement un ensemble de positions comme le montre la figure II-9; les valeurs correspondant aux positions sélectionnées dans parent 1 (respectivement dans parent 2) sont dans ce cas transmis en gardant la même position vers l'enfant 2 (respectivement l'enfant 1). Les positions restées vides dans l'enfant 2 (respectivement l'enfant 1) sont remplies avec les valeurs manquantes en les prenant dans l'ordre du parent 2 (respectivement parent 1). Ce croisement a pour avantage de prendre en compte la contrainte d'antériorité dans le processus de croisement lui-même au niveau des positions sélectionnées.

Position	1	2	3	4	5	6	7	8
Parent 1	B	D	A	E	C	G	F	H
Parent 2	A	C	D	B	F	E	H	G
Positions sélectionnées		*		*	*	*	*	
Enfant 1	D	C	A	B	F	E	H	G
Enfant 2	A	D	B	E	C	G	F	H

Figure II-9-Croisement basé sur les positions UPX

Rappelons que le problème traité est de type SHMP de type job shop faisant ainsi intervenir un grand nombre de contraintes et intégrer une complexité de résolution. Comme ce croisement

prend en considération les contraintes, il sera adopté pour présenter l'opérateur de croisement proposé lors du développement de l'algorithme génétique proposé.

- Croisement de cycle CX

Le CX (Cycle Crossover) permet de garder les positions absolues constituant les parents. Il a été développé à l'origine pour résoudre le problème du voyageur de commerce. Dans une première phase, un chromosome est construit à partir des deux parents et transmis vers l'enfant 1 (respectivement vers l'enfant 2). Dans une seconde phase, les positions vides restantes sont remplies avec les éléments du parent 2 (respectivement parent 1) pour l'enfant 1 (respectivement pour l'enfant 2).

La première phase de la construction du chromosome de l'enfant 1 (respectivement de l'enfant 2) s'effectue comme suit :

- Choisir arbitrairement une position de départ, appelée position courante
- Copier l'élément contenu dans la position courante de parent 1 (respectivement de parent 2) dans l'enfant 1 (respectivement dans l'enfant 2)
- Chercher la position de cet élément dans parent 2 (respectivement dans parent 1) cette dernière devient la nouvelle position courante.
- Si la position courante est déjà remplie dans l'enfant 1 (respectivement dans l'enfant 2), la première phase est terminée sinon revenir à la phase copiage.

II.5.2.4. Opérateur de Mutation

Cet opérateur consiste à changer aléatoirement la valeur de certains gènes dans un chromosome. On distingue deux types de mutation. La mutation à un point et la mutation d'échange ou mutation à 2 points, [Barecke, 2009].

- ***Mutation à un point***

Elle correspond à une « erreur » produite quand le chromosome est copié et reproduit pour la mutation à un point.

- ***Mutation d'échange ou mutation à 2 points***

L'opérateur de mutation classique pour les permutations est un opérateur de transposition (mutation d'échange ou swap mutation) qui échange simplement deux gènes. Le but de la mutation est de préserver la diversité dans la population et donc d'éviter la convergence prématurée.

II.5.3. L'algorithme génétique proposé

Les AGs, initialement développés pour répondre à des besoins spécifiques en biologie, ont été rapidement appliqués à la résolution avec succès de nombreux problèmes dans des différents domaines, avec des modifications selon le type du problème à traiter.

Ces modifications peuvent être dans certains cas indispensables en particulier dès lors que le problème est soumis à des contraintes très fortes qui doivent être prises en compte. Le problème d'ordonnement en est le meilleur exemple ; en effet, dès l'application de l'algorithme génétique au problème d'ordonnement traité dans ce mémoire, une modification au niveau du codage s'avère importante pour palier toutes les contraintes posées lors de la construction de l'ordonnement.

II.5.3.1. Génération de la population

La population initiale est générée par l'algorithme de satisfaction de contraintes hybridé avec SPTF, développé dans ce même chapitre. Cette approche prend en considération toutes les contraintes que présente le problème SHMP de type job shop. La procédure consiste à trouver, pour chaque opération, la cuve correspondante avec le temps de traitement minimum tout en respectant la contrainte d'antériorité. La deuxième phase de cette procédure consiste à calculer le temps d'attente de robot qui valide la faisabilité de l'ordonnement construit. Le calcul du temps du cycle est réalisé en dernière étape. L'algorithme est décrit en détails précédemment dans la section II.4.

II.5.3.2. Codage proposé

Afin de mettre en œuvre les AGs, nous avons besoin de représenter les individus symboliquement. Un chromosome est généralement une succession de binaires; ce codage proposant une représentation indirecte, ne permet pas de traiter les problèmes sur-contraints. Une représentation directe est adoptée pour le codage proposé. Chaque individu est alors paramétré

par une liste de données $(j, i, T_k, p, a_{ji}^k, b_{ji}^k, \alpha_{ji}^k)$, inspirée du codage proposé dans [Bruns, 93] pour les problèmes job shop ayant plusieurs contraintes. Ce codage fournit respectivement le numéro du produit, le numéro de l'opération, le bain relatif, la position de l'opération (l'ordre de l'opération sur la ligne), la borne minimale, la borne maximale du temps du traitement ainsi que le temps effectif de traitement à effectuer, avec :

j : $j^{\text{ème}}$ produit est le nombre d'emplois;

i : $i^{\text{ème}}$ operation;

T_k : $k^{\text{ème}}$ bain;

p : position de l'opération sur la ligne de traitement de surfaces;

a_{ji}^k : temps de traitement minimum relatif à la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit traité dans le $k^{\text{ème}}$ bain;

b_{ji}^k : temps de traitement maximum relatif à la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit traité dans le $k^{\text{ème}}$ bain;

α_{ji}^k : temps final du traitement.

La longueur d'un chromosome est égale au nombre total de tâches. En se référant aux tableaux II-9, une séquence $S = (O_{13}, T_3, p, a_{13}^3, b_{13}^3, \alpha_{13}^3), (O_{23}, T_6, p, a_{23}^6, b_{23}^6, \alpha_{23}^6), (O_{34}, T_{21}, p, a_{34}^{21}, b_{34}^{21}, \alpha_{34}^{21})$ de l'individu est représentée par le chromosome suivant (1,3,3,1,300,600,300), (2,3,6,2,60,180,60), (3,4,21,3,60,∞,60).

II.5.3.3. Opérateurs proposés

- **Opérateur de sélection**

La phase de sélection consiste à choisir parmi les N individus de la population courante les plus forts individus à partir desquels la génération suivante est créée.

Pour un ordonnancement réalisable, la valeur de la fonction fitness ou le critère d'optimisation « temps de cycle » pour le problème étudié est calculée. Une stratégie classant les individus dans un ordre croissant selon leur fonction fitness est utilisée; le plus fort des individus est celui ayant

le minimum des temps de cycle. Si on a N individus, les premiers $N/2$ individus sont sélectionnés pour la génération suivante.

- *Opérateur de croisement UPX*

Afin de respecter la contrainte d'antériorité, la permutation n'est adoptée que pour les opérations n'appartenant pas au même produit. Notons que les permutations effectuées ne génèrent pas forcément des ordonnancements faisables ; les contraintes d'intervalle ne sont plus forcément respectées puisque l'ordre des opérations et donc les temps d'attente du robot sont modifiés. Ce type de croisement peut générer parfois des enfants similaires à leurs parents.

Pour le croisement proposé, vu le nombre de contraintes à respecter, il est nécessaire de modifier ce codage pour l'adapter au problème. Pour se faire, on va supposer que le nombre des positions sélectionnées va être égal au nombre des opérations M moins 2, $(M - 2)$, c'est-à-dire que le croisement va se faire au niveau de deux opérations. Pour illustrer ce croisement dans les figures suivantes, considérons une séquence de 5 opérations

-sélectionner aléatoirement un ensemble de positions du parent 1, Fig II-10,

-produire le chromosome du premier enfant en copiant les symboles sur ces positions vers les positions correspondantes dans le chromosome de l'enfant 1, Fig II-11,

-supprimer les symboles déjà sélectionnés dans le second parent. La séquence résultante contient seulement les symboles qui vont être transmis vers l'enfant, Fig II-12,

-placer ces symboles dans des positions non fixées dans l'enfant de gauche à droite selon l'ordre de la séquence utilisée pour finir avec la construction de ce dernier, Fig II-13.

<i>Parent 1</i>	O_{11}	O_{21}	O_{32}	O_{31}	O_{22}	O_{12}	O_{13}
	$j = 1$	$j = 2$	$j = 3$	$j = 3$	$j = 2$	$j = 1$	$j = 1$
	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 2$	$i = 2$	$i = 3$
	T_1	T_1	T_4	T_1	T_2	T_9	T_3
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
	<i>300</i>	<i>300</i>	<i>60</i>	<i>300</i>	<i>60</i>	<i>60</i>	<i>600</i>
	<i>600</i>	<i>600</i>	∞	<i>600</i>	∞	∞	<i>1800</i>
	<i>300</i>	<i>300</i>	<i>60</i>	<i>300</i>	<i>60</i>	<i>60</i>	<i>600</i>
	<i>Positions sélectionnées</i>	*	*	*	*		
<i>Parent 2</i>	O_{13}	O_{32}	O_{21}	O_{11}	O_{12}	O_{31}	O_{22}
	$j = 1$	$j = 3$	$j = 2$	$j = 1$	$j = 3$	$j = 1$	$j = 2$
	$i = 3$	$i = 2$	$i = 1$	$i = 1$	$i = 1$	$i = 2$	$i = 2$
	T_3	T_4	T_1	T_1	T_9	T_1	T_2
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
	<i>600</i>	<i>60</i>	<i>300</i>	<i>300</i>	<i>60</i>	<i>300</i>	<i>60</i>
	<i>1800</i>	∞	<i>600</i>	<i>600</i>	∞	<i>600</i>	∞
	<i>600</i>	<i>60</i>	<i>300</i>	<i>300</i>	<i>60</i>	<i>300</i>	<i>60</i>

Figure II-10-Présentation des 2 parents

<i>Enfant 1</i>	O_{11}	O_{21}	O_{32}	O_{31}			O_{13}
	$j = 1$	$j = 2$	$j = 3$	$j = 3$			$j = 1$
	$i = 1$	$i = 1$	$i = 2$	$i = 1$			$i = 3$
	T_1	T_1	T_4	T_1			T_3
	1	2	3	4			7
	300	300	60	300			600
	600	600	∞	600			1800
	300	300	60	300			600
<i>Positions sélectionnées</i>	*	*	*	*			*

Figure II-11-Copie du Parent 1 vers Enfant 1

<i>Parent 2</i>	O_{13}	O_{32}	O_{21}	O_{11}	O_{12}	O_{31}	O_{22}
	$j = 1$	$j = 3$	$j = 2$	$j = 1$	$j = 1$	$j = 3$	$j = 2$
	$i = 3$	$i = 2$	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 2$
	T_3	T_4	T_1	T_1	T_9	T_1	T_2
	1	2	3	4	5	6	7
	600	60	300	300	60	300	60
	1800	∞	600	600	∞	600	∞
	600	60	300	300	60	300	60

Figure II-12-Suppression des positions sélectionnées dans Parent 2

<i>Enfant 1</i>	O_{11}	O_{21}	O_{32}	O_{31}	O_{12}	O_{22}	O_{13}
	$j = 1$	$j = 2$	$j = 3$	$j = 3$	$j = 3$	$j = 2$	$j = 1$
	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 1$	$j = 2$	$i = 3$
	T_1	T_1	T_4	T_1	T_9	T_2	T_3
	1	2	3	4	5	6	7
	300	300	60	300	60	60	300
<i>Positions sélectionnées</i>	*	*	*	*			*

Figure II-13-Remplissage des trous de l'enfant 1

En procédant de la même façon, la construction de l'enfant 2 peut être obtenue. Pour le cas étudié, si les opérations restantes dans le parent 2 appartiennent au même produit et le croisement va inverser l'ordre de ces opérations, il est inutile de continuer avec l'enfant, car, cet ordonnancement ne respectant pas la contrainte d'antériorité n'est pas faisable, Fig II-14.

<i>Parent 1</i>	O_{11}	O_{21}	O_{32}	O_{31}	O_{22}	O_{12}	O_{13}
	$j = 1$	$j = 2$	$j = 3$	$j = 3$	$j = 2$	$j = 1$	$j = 1$
	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 2$	$i = 2$	$i = 3$
	T_1	T_1	T_4	T_1	T_2	T_9	T_3
	1	2	3	4	5	6	7
	300	300	60	300	60	60	300
	<i>Positions sélectionnées</i>	*	*	*	*	*	
<i>Parent 2</i>	O_{13}	O_{32}	O_{21}	O_{11}	O_{12}	O_{31}	O_{22}
	$j = 1$	$j = 3$	$j = 2$	$j = 1$	$j = 3$	$j = 1$	$j = 2$
	$i = 3$	$i = 2$	$i = 1$	$i = 1$	$i = 1$	$i = 2$	$i = 2$
	T_3	T_4	T_1	T_1	T_9	T_1	T_2
	1	2	3	4	5	6	7
	300	60	300	300	60	300	60
	<i>Enfant 1</i>	O_{11}	O_{21}	O_{32}	O_{31}	O_{22}	O_{13}

Figure II-14-Cas d'un ordonnancement non faisable

- *Opérateur de mutation à un point*

Cet opérateur consiste à choisir aléatoirement une opération et à changer le numéro de la cuve relative, comme le montre la figure II-15. Puisque le temps de déplacement intervient dans le calcul du temps du cycle, une mise à jour est ensuite effectuée conduisant à une modification de la fonction fitness.

<i>Parent</i>	O_{13}	O_{32}	O_{21}	O_{11}	O_{12}	O_{31}	O_{22}
	$j = 1$	$j = 3$	$j = 2$	$j = 1$	$j = 1$	$j = 3$	$j = 2$
	$i = 3$	$i = 2$	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 2$
	T_3	T_4	T_1	T_1	T_9	T_1	T_2
	1	2	3	4	5	6	7
	300	60	300	300	60	300	60
	<i>Enfant</i>	O_{13}	O_{32}	O_{21}	O_{11}	O_{12}	O_{31}
$j = 1$		$j = 3$	$j = 2$	$j = 1$	$j = 1$	$j = 3$	$j = 2$
$i = 3$		$i = 2$	$i = 1$	$i = 1$	$i = 2$	$i = 1$	$i = 2$
T_3		T_4	T_1	T_1	T_7	T_1	T_2
1		2	3	4	5	6	7
300		60	300	300	60	300	60

Figure II-15-Fonctionnement de l'opérateur de mutation à un point

II.5.3.4. Description de l'algorithme

Les étapes décrites ci-après, expliquent le fonctionnement des différents opérateurs de l'algorithme génétique appliqué au problème d'ordonnancement traité.

Phase de génération de la population initiale

Début
Fixer la taille P de la population initiale (P étant paire)
Générer une première solution
K=1
{ Tant que (K ≠ P) Faire
 Tester la réalisabilité
 { Si (Solution respecte toutes les contraintes) alors
 Solution acceptée k=k+1
 Sinon
 Solution refusée
 Fin Si
 Générer une autre solution
 Fin tant que
Fin

Phase de la sélection

Début
{ Pour i= 1 jusqu'à P Faire
 Calculer le Temps de Cycle-Individu
 Placer les individus dans un tableau de P colonnes
 Trier les individus dans un ordre croissant
 Sélectionner de 1 jusqu'au P/2 individu
 Fin pour
Fin

Phase de la mutation

Début

Sélectionner aléatoirement un individu i

Sélectionner une opération O_{ji} qui correspond à une position k parmi M

{ Pour p allant de 1 à M Faire

{ Si $p \neq k$ alors

 Individu muté reçoit les mêmes affectations que Individu sélectionné

 Sinon

 Changer la cuve attribuée à l'opération O_{ji} au niveau de la position k

{ Fin Si

{ Fin Pour

Tester la réalisabilité

{ Si l'individu muté respecte toutes les contraintes alors

 Individu muté accepté

 Sinon

 Individu muté refusé

{ Fin Si

Fin

Phase du croisement

Début

Sélectionner aléatoirement Parent 1 et Parent 2

Sélectionner aléatoirement un ensemble de position de croisement

(pour le cas traité, le nombre de positions est égal à 2)

Pour $i, i = 1 \text{ à } M$ Faire

{ Si la position de i du parent 2 coïncide avec une position sélectionnée alors

 Enfant 1 reçoit l'opération du parent 2 dans la même position

 Supprimer cette opération dans Parent 1

Fin Si

{ Si la position de i du parent 1 coïncide avec une position sélectionnée alors

 Enfant 2 reçoit cette opération dans la même position

 Supprimer cette opération dans Parent 2

Fin Si

Fin Pour

{ Les positions vides de l'Enfant 1 reçoivent les positions restées dans Parent 1 dans l'ordre

Les positions vides de l'Enfant 2 reçoivent les positions restées dans Parent 2 dans l'ordre

Tester la réalisabilité

{ Si Enfant 1, Enfant 2 respectent toutes les contraintes alors

 Enfant 1, Enfant 2 acceptés

 Sinon

 Enfant 1, Enfant 2 pas acceptés

Fin Si

Fin

Après la phase de permutation et croisement, la faisabilité du nouvel ordonnancement est testée, conformément au nouvel ordre donné. Pour ce fait, il est d'abord nécessaire de vérifier la contrainte de capacité de la cuve. Rappelons que $Indicateur = [I_1, I_2, \dots, I_N]$ est le vecteur d'occupation des cuves, $I_k = 1$ pour une cuve occupée, $I_k = 0$ pour une cuve vide. Supposons que la première position dans la permutation est k , la deuxième position est $k+1$, $k = 1, \dots, M-1$ avec M étant le nombre total des opérations. Par conséquent, une permutation est possible si $I_{k+1} = 0$ pour la $k^{ème}$ itération. Dans le cas de permutation non faisable, on a $I_{k+1} = 1$.

Le temps des déplacements et les temps d'attente du robot sont ensuite recalculés. Si le nouveau planning des tâches satisfait à toutes les contraintes précédemment citées, le temps de cycle est recalculé. Si le nouveau temps cycle est supérieur à la durée du cycle initial, la solution relative est supprimée sinon elle est sauvegardée. Les ordonnancements sont produits similairement jusqu'à atteindre une valeur optimale du temps de cycle.

II.5.3.5. Résultats et simulation

Ainsi, à partir d'une population initiale générée par l'application de l'algorithme de satisfaction de contraintes hybridé avec SPTF, les algorithmes génétiques sont utilisés afin de résoudre le problème d'ordonnancement en vue d'obtenir une meilleure solution qui minimise une fonction de fitness (temps du cycle).

En effet, pour 1000 générations, les opérateurs de mutation et de croisement sont utilisés, générant ainsi, des nouveaux individus pour en choisir celui ayant la meilleure fonction fitness avec un temps de cycle égale à 5561.1s comme illustré dans la figure II-16.

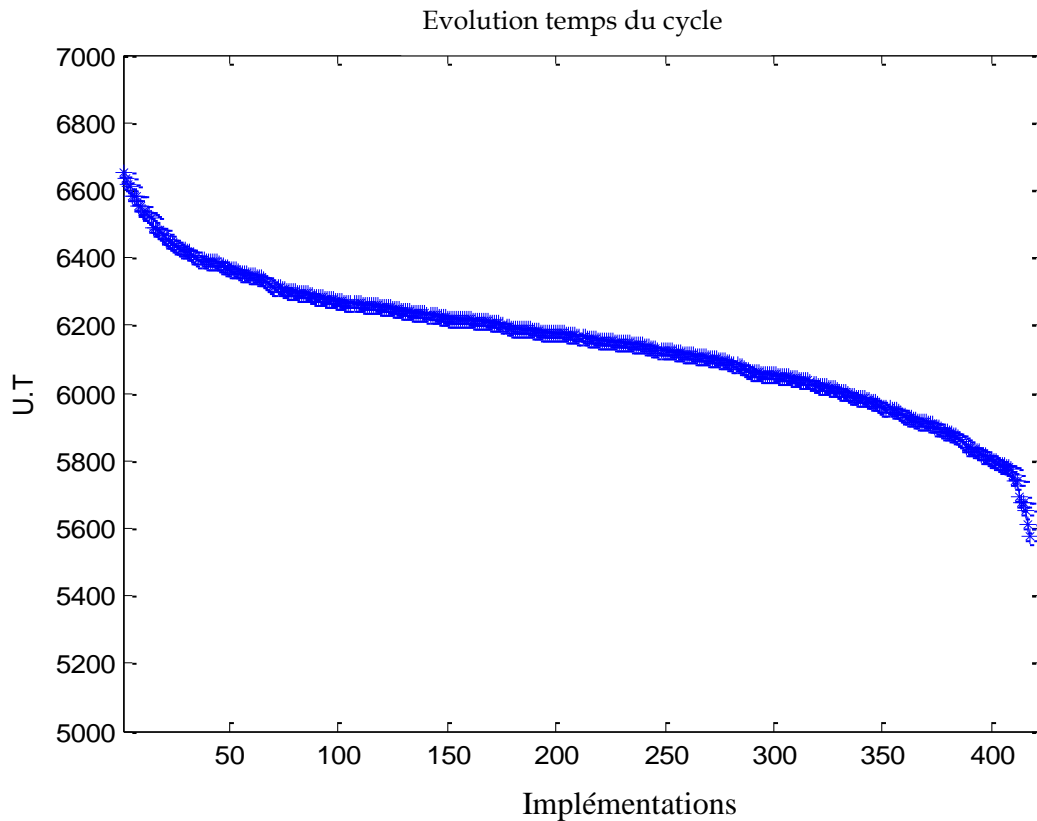


Figure II-16-Evolution du temps de cycle à travers les générations pour le problème d’ordonnancement associé à l’atelier de traitement de surfaces étudié

L’algorithme génétique proposé a été ainsi appliqué avec succès pour résoudre le problème SHMP étudié. Cette approche utilise un codage différent afin de présenter la complexité de l’atelier étudié. Par rapport à d’autres algorithmes [Yih, 94], [Lim, 97], [Pezzella et al, 08] où les temps des déplacements du chariot ne sont pas considérés ou supposés constants, une amélioration est obtenue dès que les temps de transferts du robot sont pris en compte pour l’obtention d’un ordonnancement meilleur.

Le séquençage des opérations, les cuves associées, les temps de traitement, les nouveaux temps de traitement, présentant la data de la solution obtenue, sont donnés dans le tableau II-17.

Tableau II-17-Data-Ordonnancement obtenu par application de l'AG

Opération	Bain	P_{ji}^k	α_{ji}^k
O_{13}	T_3	300	300
O_{34}	T_{21}	60	60
O_{14}	T_{12}	60	343
O_{35}	T_8	60	247.5
O_{15}	T_5	60	135.6
O_{23}	T_6	60	60
O_{24}	T_5	60	60
O_{36}	T_{22}	60	362.8
O_{25}	T_{17}	60	212.9
O_{37}	T_{13}	60	180
O_{16}	T_6	60	60
O_{17}	T_{12}	60	60
O_{18}	T_{16}	1800	1800
O_{26}	T_8	60	60
O_{38}	T_{12}	60	60
O_{19}	T_4	60	97.8
O_{27}	T_9	60	187.3
$O_{1,10}$	T_{17}	60	60
O_{28}	T_{10}	1800	1800

O_{39}	T_9	60	60
O_{29}	T_{11}	60	60
$O_{3,10}$	T_{23}	60	88.8
$O_{1,11}$	T_{18}	60	60
$O_{2,10}$	T_7	60	295.6
$O_{1,12}$	T_{19}	60	60
$O_{1,13}$	T_{20}	60	60
O_{31}	T_1	60	63.5
$O_{1,14}$	T_4	60	60
O_{32}	T_{22}	60	60
$O_{2,11}$	T_{23}	60	60
$O_{1,15}$	T_9	60	60
O_{21}	T_1	60	300
$O_{1,16}$	T_{23}	60	60
O_{11}	T_1	60	300
O_{33}	T_5	60	60
O_{12}	T_{21}	60	60
O_{22}	T_{22}	60	60

II.6. Conclusion

La résolution des problèmes d'ordonnancement et en particulier dans le cas des problèmes SHMP de type job shop avec intégration des temps de déplacements du chariot effectifs, constitue la contribution principale de ce chapitre. Après avoir introduit un atelier de traitement de surfaces réel de type job shop, une démarche globale de résolution de ce problème a été proposée. Elle est constituée de 2 étapes : une approche exacte est développée en première phase suivie d'une analyse méta-heuristique en deuxième phase.

Un algorithme de satisfaction des contraintes palliant toutes les contraintes faisant la complexité de l'atelier étudié est développé pour donner la naissance à un nombre de solutions limité. Une hybridation avec SPTF est introduite ensuite a permis l'amélioration des résultats obtenus. Cette approche, combinant deux heuristiques, a généré la population initiale nécessaire à l'application des algorithmes génétiques qui ont fait sa preuve dans la recherche d'un minimum global et ont permis d'atteindre une solution de meilleure qualité. L'évaluation des solutions obtenues lors de la mise en œuvre en ligne nécessite de mener une étude de la robustesse de ces solutions en présence des perturbations. La démarche de l'étude de la robustesse est envisagée dans le chapitre suivant.

Chapitre III : Approche proposée pour la caractérisation de la robustesse d'un ordonnancement d'ateliers de traitement de surfaces

III.1. Introduction

Les problèmes d'ordonnancement sont généralement résolus en générant une solution faisable suite à l'application d'une méthode exacte ou approchée en proposant une ou plusieurs solutions bien adaptées.

Les heuristiques et les métaheuristiques, discutées dans le chapitre précédent, conduisent parfois à des solutions sensibles aux perturbations lors de la mise en ligne. Ce problème s'avère, en effet, d'une grande importance dans les cas des problèmes d'ordonnancement réels et nécessite la prise en compte des perturbations et la détermination d'un voisinage de la solution obtenue théoriquement garantissant son efficacité lors de sa mise en œuvre.

Le problème, étudié dans ce chapitre, concerne ainsi la définition, l'évaluation et l'intégration de la robustesse dès la phase initiale de la résolution d'un problème d'ordonnancement.

L'idée est de prendre en considération ces incertitudes, de les incorporer dès la formulation du problème, de mesurer la sensibilité de l'ordonnancement construit et d'améliorer celle-ci avant de passer à la mise en œuvre en pratique.

Pour se faire, une approche par scénarii basée sur l'étude du cas industriel d'une ligne de traitement de surface, présenté dans le chapitre précédent, est développée dans la première partie du présent chapitre.

Inspirée de celle proposée par Rossi, [Rossi, 03], elle garantit des performances en présence des perturbations. Les travaux consignés [Herroelen et al, 02], confirme cette approche, dont

l'utilisation est aussi illustrée dans [Kouvelis et al, 00], ou encore dans [Vincke, 99a] et [Vincke, 99b]. Parmi les méthodes d'évaluation multi-critères des indicateurs de performance d'une solution robuste, on cite principalement : la méthode d'agrégation, la méthode de compromis et la méthode de programmation par but [Talbi, 99], [Collette et al, 02].

Une méthode d'optimisation multi-critères, utilisant l'agrégation par l'intégrale de Choquet, est proposée dans ce chapitre pour être mise en oeuvre. Elle consiste à générer une variété de solutions dans un espace de recherche, et à aider le décideur quand il ne peut pas donner une préférence particulière à l'une des fonctions objectif.

Un algorithme permettant de déterminer d'une manière systématique la robustesse d'une solution en effectuant une évaluation multi-critères est développé à la fin avec ce chapitre.

III.2. Robustesse, sensibilité et flexibilité pour les problèmes d'ordonnancement

III.2.1. Historique

Le terme robustesse paraît être bien établi dans la littérature, tandis que la flexibilité n'a été introduite que récemment dans [Branke et al, 00] et [Jensen, 00]. Malgré l'utilisation fréquente de la robustesse, sa signification reste néanmoins différente d'un auteur à un autre. La vérification de la propriété de robustesse est une préoccupation de plus en plus présente dans les domaines de la recherche opérationnelle et d'aide à la décision où elle donne lieu à des démarches et des concepts de plus en plus riches et diversifiés. Il s'agit de modéliser les imprévus et d'en tenir compte lors de la prise de la décision. En effet, les incertitudes ne sont pas de simples déviations temporelles de données, mais font partie du futur. La meilleure manière de manipuler les incertitudes est de les modéliser et les considérer à part entière dans les modèles de résolution. L'objectif alors est de trouver des solutions dont les performances sont peu sensibles aux perturbations sur les données qui peuvent survenir. Plusieurs définitions de la robustesse sont données, selon le contexte et le problème posé.

Dans la littérature, les auteurs ont des définitions différentes de la robustesse. Ainsi dans le

domaine des statistiques, [Huber, 81], la robustesse est considérée comme étant une insensibilité à toutes déviations par rapport aux hypothèses. En recherche opérationnelle et plus particulièrement dans le contexte d'aide à la décision, Roy, est proposé le concept de conclusion robuste qui vise à élaborer des éléments de réponse à un problème auquel est confronté un décideur, éléments de réponses qui sont obtenus en prenant en compte les incertitudes éventuelles sur les paramètres du problème [Roy, 97]. Dans la théorie de la décision, l'action la plus robuste est celle qui ne conduit pas à une perte importante quelque soit l'événement qui va se produire, [Pomerol, 01]. D'un point de vue de la conception de produits, la notion de robustesse fait référence à l'insensibilité relative des performances fonctionnelles d'un produit, [Kackar, 85].

III.2.2. Terminologie

III.2.2.1. Flexibilité

L'introduction de la flexibilité dans un problème d'ordonnancement reflète le degré d'adaptation pendant la phase de sa mise en œuvre tout en respectant les contraintes posées. Cette flexibilité peut toucher plusieurs niveaux:

- le temps, ou la flexibilité temporelle, relatif aux dates de début des opérations constituant le premier niveau de flexibilité dans la planification,
- l'ordre d'exécution, ou la flexibilité séquentielle, relatif à la capacité de changer l'ordre dans lequel les opérations doivent fonctionner sur les machines, supposant implicitement la flexibilité temporelle. Pendant l'exécution d'une séquence, certaines opérations peuvent ainsi dépasser d'autres, si les conditions l'exigent,
- les affectations, en présence de ressources multiples, permettant à une tâche d'être exécutée en utilisant une ressource autre que celle qui était initialement prévue. Cette flexibilité, très utile, par exemple quand une machine devient indisponible, suppose implicitement la flexibilité séquentielle et la flexibilité temporelle,
- le mode d'exécution comprenant les possibilités de préemption dans le cas où deux opérations se chevauchent et peuvent être proposé s'il s'agit de surmonter une situation difficile sauf si cette notion de non préemption est impérative.

III.2.2.2. Robustesse

Définition 1

La robustesse d'un système peut être définie comme sa capacité à garder (conserver) les propriétés qualifiant ce système soumis à des perturbations prévues ou imprévues.

La robustesse se définit à l'égard du type des perturbations et de la qualité requise à la sortie du système [Jerbi, 06].

Pour les systèmes de production industriels, parmi les critères de sortie, on distingue le respect de la gamme, la conservation du flux de production, les temps de séjour et le temps de cycle dans l'atelier. Parler de la robustesse d'un système revient à mesurer sa résistance à conserver ses propriétés face à une perturbation définie à l'avance [Jerbi, 06].

Pour chaque système et chaque niveau, la robustesse peut être de deux types différents, [Collart-Dutilleul, 92] :

- la robustesse interne qui concerne les changements de valeur de paramètres du modèle du procédé.

- la robustesse externe qui touche, quant à elle, les variations acceptables à l'entrée du système.

Pour une cellule de production, les perturbations externes sont, par exemple, les variations des gammes de fabrication, des taux de production pour une même gamme ou les variations des instants d'arrivée des produits. La robustesse externe de la cellule qualifie la capacité à changer d'objectifs (gamme et/ou taux de production) et à les atteindre indépendamment des fluctuations des entrées. Les perturbations internes sont, par exemple, les variations des temps opératoires ou les pannes des machines.

Selon la nature de la réaction face à une perturbation, la robustesse peut être aussi décomposée en deux classes : robustesse passive et robustesse active [Collart-Dutilleul, 92].

Définition 2

La robustesse passive répond au cas où aucune modification dans la conduite n'est nécessaire pour que les propriétés spécifiées soient conservées en présence de variations.

Définition 3

La robustesse active correspond au cas où les propriétés spécifiées peuvent être maintenues, mais au prix d'un calcul total ou partiel de la conduite.

III.2.2.3. Flexibilité et robustesse

La robustesse veut dire que l'ordonnancement reste acceptable si des aléas perturbent l'exécution de l'ordonnancement, tandis que la flexibilité signifie l'aptitude d'introduire des changements dans l'ordonnancement.

Dans la plupart des applications de l'ordonnancement (gestion d'ateliers, parallélisme, gestion de projets), il existe toujours une incertitude plus ou moins grande sur les données associées au problème considéré et sur la robustesse du système étudié [Tagina et al, 09]. Il peut s'agir de la durée des tâches, mais aussi de la disponibilité des ressources.

L'incertitude n'est alors levée que lors de l'exécution effective des tâches. L'approche déterministe est inadaptée à des problèmes où l'ampleur de l'incertitude est trop importante alors que l'approche stochastique postule une connaissance en fait très fine de l'incertitude (modèle aléatoire), mais se heurte à de grosses difficultés calculatoires.

III.2.3. Classification des perturbations

Une perturbation est définie, d'une manière générale, comme tout événement opérationnel non prévu pouvant affecter les performances d'un ordonnancement généré, dans un environnement statique où toutes les données du problème sont connues à l'avance et aucun changement dans l'environnement n'est envisagé. La perturbation intervient lors de la mise en œuvre de ce dernier. Selon Baillet, [Baillet, 94], les perturbations peuvent être internes ou externes au système de production.

-Les perturbations internes sont des événements opérationnels imprévus qui se produisent au sein du système de production. Elles correspondent, par exemple, à un prolongement de la durée d'une ou de plusieurs tâches, à une panne machine, à une absence inopinée d'un opérateur ou à une erreur d'information.

-Les perturbations externes sont des événements affectant les entrées et les sorties du système de production. On distingue alors les perturbations dues aux fournisseurs, comme la variation des

dates de disponibilité de la matière première, et celles dues aux clients qui passent et modifient fréquemment leurs commandes, au dernier moment, au détriment de celles planifiées à l'avance.

Selon [Artigues et al, 02], la classification des perturbations dépend du niveau de connaissance de l'incertitude. Les auteurs ont identifié trois types d'incertitudes dans les environnements manufacturiers.

-Les incertitudes complètement inconnues, ou bien aléas imprévisibles, sont des événements imprévus pour lesquels aucune information n'est disponible à l'avance. C'est le cas d'une grève, un accident dans le lieu de travail ou une absence inopinée d'un ouvrier, etc, dont il est impossible d'en tenir compte dans la phase de pré calcul d'un ordonnancement prédictif initial.

-Les suspicions du futur sont formulées grâce à l'intuition et l'expérience du décideur. Elles constituent des incertitudes difficilement quantifiables et donc difficiles à intégrer dans des algorithmes d'ordonnancement. Des systèmes d'aide à la décision interactifs sont plus appropriés pour ce type d'incertitudes.

-Les incertitudes partiellement connues, ou bien aléas prévisibles, sont telles que seules quelques informations sont disponibles avant l'exécution des opérations. On peut, par exemple, connaître le type de perturbations qui peuvent arriver (pannes machines, arrivée des tâches en retard, etc.). Un plus d'informations peut être déduit grâce à l'historique de l'atelier, comme par exemple, les lois de distribution de la fréquence d'occurrence et de la durée des pannes. Ces informations peuvent être incluses lors du calcul d'une solution initiale au problème considéré.

Les données associées à un problème d'ordonnancement sont les durées, les dates d'occurrence de certains événements, certaines caractéristiques structurelles, et les coûts.

Aucune de ces données n'est exempte de facteurs d'incertitudes. Dans [Mehta et al, 99] et dans [Cowling et al, 02], les auteurs ont considéré essentiellement deux classes d'incertitudes : celles liées aux ressources et celles liées aux travaux que nous envisageons de définir dans la section suivante.

III.2.3.1. Perturbations sur les ressources

Les perturbations prises en compte dans cette catégorie sont les pannes machines, le changement de ressources et le manque de ressources.

-Les pannes machines peuvent suivre des lois de probabilité ou être complètement aléatoires. On peut inclure aussi, dans cette catégorie, le facteur humain, l'opérateur travaillant sur la machine [Alcaide et al, 02], [Lawrence et al, 97] (i.e. on peut considérer l'indisponibilité d'un ou de plusieurs opérateurs comme une panne machine, car la machine commandée par cet opérateur reste indisponible). La durée de la panne d'une machine peut être ainsi vue comme la durée de l'indisponibilité de l'opérateur. Les pannes machines peuvent être classées par niveaux, chaque niveau déterminant une grandeur des pannes envisagées pour ce niveau (le nombre de pannes dans l'atelier ou le nombre de machines affectées).

Cette classification est effectuée pour mieux étudier le comportement de l'ordonnancement et évaluer la robustesse de ce dernier pour chaque niveau en présence [Shafaei et al, 99a], [Jensen, 01], [Pistikopoulos et al, 96] et [Balasubramanian et al, 02].

-Le changement de ressources concerne la modification dans les matières premières pour la fabrication d'un produit ou le changement d'un ou de plusieurs outils d'une machine et ceci dans le but d'accomplir une autre tâche par la même machine [Mignon et al, 95] et [Lawrence et al, 97].

-Le manque de ressources concerne soit le manque de matières premières ou d'outils de fabrication. Ce type de perturbations est rencontré dans les ateliers ne disposant pas de stock, ou les ateliers n'ayant pas une livraison adéquate et régulière, [Lawrence et al, 97].

III.2.3.2. Perturbations sur les travaux

Parmi les perturbations les plus fréquentes et les plus étudiées, on peut trouver :

-l'ajout d'un nouveau travail : dans les situations réelles, on peut inclure des commandes (travaux) urgentes et ceci pendant l'exécution de l'ordonnancement;

-l'annulation d'un travail : elle se produit quand un travail n'est plus nécessaire dans le déroulement de l'atelier [Lawrence et al, 97] ;

-le changement de la priorité dans les travaux : cette situation est rencontrée seulement si les travaux n'ont pas la même importance. Un changement dans leur ordre de priorité conduit donc inévitablement à un changement dans les affectations des ressources et l'ordre de passage des travaux par les machines [Mignon et al, 95];

-la charge d'un atelier : elle correspond au degré de liberté d'exécution de chaque tâche dans l'atelier. On dit qu'un atelier est chargé si la tâche n'a pas beaucoup de liberté d'exécution ;

-la modification d'un travail : en général, cette modification concerne le changement dans les durées opératoires des différentes tâches d'un travail ou d'un ensemble de travaux (élargissement ou rétrécissement de la durée opératoire), [Lawrence et al, 97], [Shafaei et al, 99a].

L'ensemble des auteurs, qui ont parlé des incertitudes dans les ordonnancements, ont étudié principalement deux perturbations : perturbations sur les équipements et plus particulièrement les pannes machine ou encore les perturbations concernant les travaux et plus particulièrement le changement dans les durées opératoires. L'étude proposée dans ce chapitre concerne ces deux types de perturbation du fait qu'une perturbation au niveau des ressources (le chariot) entraîne directement une perturbation au niveau des travaux (durées opératoires des différentes tâches).

Un aperçu sur les méthodes d'ordonnement de la production en présence de perturbations est présenté dans la section suivante.

III.2.4. Classification des méthodes d'ordonnement en présence de perturbations

Plusieurs approches d'ordonnement et de réordonnement ont été proposées dans la littérature pour tenir compte de la présence de données incertaines et des perturbations qui peuvent survenir en temps réel dans l'atelier [Benbouzid, 05]. Mehta et Uzsoy, , proposent une classification de ces approches en quatre catégories : les approches totalement réactives, les approches prédictives réactives, les approches robustes et les approches à base de connaissances. Davenport et Beck proposent une deuxième classification en distinguant les approches réactives, les approches proactives et les approches mixtes [Mehta et al, 99], [Davenport et al, 00].

Les approches réactives [Davenport et Bech, 00] se basent sur des informations mises à jour en temps réel selon l'état du système et éventuellement sur un ordonnancement initial calculé a priori sans intégrer de connaissances sur ce qui va se passer dans l'atelier. Ces approches correspondent exactement aux approches totalement réactives et les approches prédictives réactives [Mehta et Uzsoy, 99]. En outre, la définition des approches proactives donnée par Davenport et Beck correspond exactement à celle donnée aux approches robustes par Mehta et Uzsoy. Cependant, la classification de Mehta et Uzsoy ne permet pas de classer les approches mixtes de Davenport et Beck.

Les approches à base de connaissances peuvent être considérées comme des approches réactives où l'algorithme réactif consiste à sélectionner, chaque fois qu'une décision doit être prise, une règle de réordonnancement ou de séquençement parmi un ensemble d'alternatives selon l'état de l'atelier et les connaissances acquises dans des expériences antérieures.

Dans la suite du présent chapitre, nous adopterons la classification de Davenport et Beck.

III.2.4.1. Les approches réactives

On distingue les approches totalement réactives, ou réactives pures, qui n'utilisent pas d'ordonnancement prédictif initial et les approches prédictives réactives qui utilisent à la fois un algorithme prédictif qui calcule un ordonnancement initial hors ligne, sans tenir compte de la présence de perturbations, et un algorithme réactif qui tient compte du nouvel état du système pour recalculer une nouvelle solution chaque fois qu'un aléa survient.

Dans l'approche réactive pure, aucun ordonnancement initial n'est donné à l'atelier. Les décisions sont prises localement en utilisant, dans la plupart des cas, des règles de priorité. Dans les approches prédictives réactives, un ordonnancement prédictif est généré initialement sans tenir compte de futurs aléas.

III.2.4.2. Les approches proactives ou robustes

Contrairement aux approches réactives, dans les approches proactives ou robustes, la connaissance de l'incertitude est utilisée par l'algorithme statique pour construire un seul ordonnancement (ordonnancement de référence, en anglais baseline schedule) ou une famille d'ordonnancements. On trouve aussi dans la littérature le terme d'approche prédictive, la

différence étant que l'ordonnancement construit par un algorithme statique dans ce cas, ne tient pas compte des incertitudes.

Le but de l'approche proactive est de prendre en considération les incertitudes lors de la mise en œuvre de l'ordonnancement prédictif original. Cette prise en compte des informations incertaines est utile pour rendre l'ordonnancement prédictif plus robuste. Selon cette approche un ordonnancement robuste est caractérisé par la possibilité de rester valide sous une large variété de perturbations.

III.2.4.3. Les approches proactives réactives

« Un système d'ordonnancement capable de gérer les incertitudes utilise vraisemblablement non seulement une approche proactive mais aussi une approche réactive » [Davenport et al, 00].

Il est naturel de coupler une approche proactive, lorsqu'elle propose une famille d'ordonnancements, avec une phase dynamique élaborée : au fur et à mesure de la connaissance des valeurs réelles des données et éventuellement suite à un aléa, un algorithme dynamique est utilisé pour choisir parmi les ordonnancements préalablement sélectionnés en statique, ceux ayant les meilleures performances. Cette approche, qui permet de réagir aux conditions réelles, est appelée approche proactive/réactive.

En effet, il est très difficile de tenir compte proactivement de toutes les perturbations qui surviennent réellement, à part peut être le cas du sacrifice des performances de l'ordonnancement généré hors ligne en insertion par exemple d'un très grand nombre de temps morts évidemment non souhaités.

III.3. Modèle pour l'étude de la robustesse pour le problème SHMP

Dans cette section, nous présentons d'une façon globale le modèle que nous proposons pour l'étude de la robustesse des ordonnancements en présence de perturbations ainsi que des mesures de robustesse déterminant les indicateurs des performances d'un ordonnancement.

III.3.1. Contexte de l'étude

Un ensemble de produits est traité dans la chaîne de traitement de surfaces qui, schématiquement, comporte, une séquence de bains présentant chacun un traitement bien spécifié de l'atelier a détaillé au paragraphe II.2.

Dans ce mémoire, deux cas peuvent se présenter.

- chaque traitement est délimité par des temps opératoires qui sont la borne minimale et la borne maximale ;
- les différents traitements ne sont pas bornés par une borne maximale.

L'ordonnement doit de plus tenir compte des contraintes techniques et des considérations économiques.

- Les contraintes techniques d'ordonnement résultent principalement de la limitation du temps opératoire de certains postes de la ligne de traitement de surfaces par une marge maximale et une autre minimale. Ces contraintes deviennent plus dures dans le cas où des temps de séjour fixes sont spécifiés pour chaque poste.

Pour ces postes, qualifiés de critiques au niveau des temps opératoires, une perturbation extérieure peut conduire au dépassement de l'une des marges maximales introduisant l'endommagement du produit en cours de traitement et rend l'ordonnement non faisable. Concrètement, cela se traduit par une prise en compte de contraintes d'espacement entre le temps de séjour de départ et la marge maximale pour pouvoir compenser les perturbations extérieures sans toucher à la faisabilité de l'ordonnement.

Pour des raisons économiques, on cherche souvent à choisir des temps de séjour proches des bornes minimales. Dans le cas d'une perturbation, générer un coût supplémentaire est considéré meilleur qu'endommager les pièces.

Le système productif étudié est calibré pour pouvoir travailler :

- avec des temps opératoires bornés entre des marges minimales et des marges maximales ;
- avec un temps de cycle donné ou une certaine plage de temps de cycle.

Comme l'étude envisagée porte sur la robustesse des ordonnancements dans le cadre statique, nous ne considérons que le cas hors ligne pour le modèle global de résolution.

Il est restreint à la génération d'un ordonnancement robuste à mettre en œuvre en ligne, l'idée étant de fournir, à la partie en ligne (réactif) un ordonnancement qui peut ne pas être l'optimum mais qui possède une certaine capacité à absorber les aléas.

III.3.2. Présentation globale de l'approche

L'étude envisagée sur le plan de la robustesse du système de production industriel en présence de perturbations, est constitué de quatre phases : Scénarii, Perturbations, Classification et Validation.

- Scénarii : nous distinguons quatre scénarii d'ordonnements générés suite à l'application de l'algorithme de satisfaction de contraintes développé dans le chapitre précédent, chaque scénario étant spécifié par son temps de cycle.

- Perturbation : cette phase consiste à injecter une perturbation au niveau de chaque scénario. Une moyenne de 20 simulations est effectuée pour étudier l'effet de la perturbation injectée sur chaque scénario. Durant chaque processus de perturbation, sont calculées les mesures de robustesse qui seront définies ultérieurement.

- Classification : grâce aux scénarii établis, un classement des ordonnancements selon les mesures trouvées est réalisé. Ceci permet d'identifier les indicateurs de performances permettant de qualifier la robustesse des ordonnancements, à partir des mesures initiales.

- Validation : cette phase a pour objectif de valider les conclusions de la phase précédente.

Cette validation se fait comme suit : pour les mêmes échantillons d'ordonnements de la phase précédente, un nouveau classement se basant sur les indicateurs de performances, sans passer par la phase scénarii et perturbation, est effectué.

Si nous aboutissons aux mêmes résultats, alors nous pourrions affirmer que ces indicateurs sont capables de distinguer les ordonnancements robustes. L'objectif de ce mémoire ne concernera pas la recherche d'une solution optimale générée par la métaheuristique développée dans le chapitre II, mais de déterminer une solution optimale et robuste.

Dans ce qui suit, nous définissons en particulier les mesures de la robustesse, retenues dans notre problème, ainsi que les perturbations utilisées dans la phase perturbation du modèle.

Puis, après la présentation de l'approche scénario que nous avons développée en suivant la méthodologie présentée dans cette section, nous terminons le chapitre par l'introduction des indicateurs de performances et la validation des résultats trouvés, en proposant un algorithme capable de faire la distinction d'une solution parmi un ensemble de solutions faisables.

III.3.3. Eléments de la robustesse

L'étude de la robustesse proposée nécessite de déterminer un ensemble de 3 éléments : les caractéristiques qui doivent être respectées, les perturbations injectées et enfin les mesures de robustesse proposées.

III.3.3.1. Caractéristiques

Les temps opératoires

Le type d'atelier étudié fait appel à des procédés dont les temps opératoires doivent être compris entre des marges minimales et des marges maximales. Les durées des opérations sont alors

spécifiées par des intervalles de temps. Les contraintes de temps fixent un critère d'admissibilité strict. Par ailleurs, en présence des perturbations, il devient utile de mettre en œuvre la robustesse de la solution obtenue.

Nous commençons par traiter le cas de l'atelier de traitement de surfaces décrit dans le chapitre précédent. Après le choix d'une méthode de résolution consistant à appliquer l'algorithme de satisfaction de contraintes, une solution est générée.

Lorsque les durées des opérations à effectuer sont parfaitement définies, une marge de réglage au niveau des marges maximales caractérisant la robustesse reste disponible. En effet, la qualité des produits impose souvent que les temps opératoires soient respectés de façon à ce que la borne maximale ne soit pas dépassée pour ne pas endommager les produits.

Temps d'attente

Comme décrit dans le chapitre précédent, l'algorithme de satisfaction de contraintes développé est fondé sur le calcul du temps d'attente en liaison directe avec les marges opératoires; en effet, un ordonnancement n'est faisable que si les temps d'attente relatifs à toutes les opérations sont

positifs. De ce fait, toute perturbation sur les temps opératoires entraîne des modifications sur les valeurs des temps d'attente ; un test sur la faisabilité de l'ordonnancement est à appliquer à chaque changement pour vérifier que les propriétés soient conservées.

III.3.3.2. Perturbations

Etant donné que la ressource critique dans le problème traité est le robot, il est judicieux d'étudier d'abord la robustesse de l'ordonnancement

- en injectant une perturbation au niveau de cette ressource en réduisant sa vitesse initiale,
- en étudiant la résistance à face à cette dégradation de vitesse, si l'ordonnancement est obtenu dans les conditions normales, la vitesse de décrochage étant celle qui présente la limite d'un ordonnancement faisable.

En seconde phase, il est intéressant d'étudier l'effet d'un deuxième type de perturbations agissant sur les fenêtres de temps de traitement présentant la spécificité des lignes de traitement de surfaces. Ceci peut être effectué par la diminution du niveau des marges maximales.

L'influence de cette action sur la robustesse des ordonnancements est aussi considérée pour la même perturbation sur la vitesse du chariot.

III.3.3.3. Mesures de robustesse proposées

Le critère de performance retenu ici est de conserver la faisabilité de l'ordonnancement solution, la stabilité du temps de cycle ou sa minimisation n'étant pas recherchée à ce stade.

Pour un ordonnancement donné (une solution particulière), nous introduisons une perturbation et essayons de voir à quel stade cet ordonnancement peut résister et garder ses spécificités tout en augmentant le flux de la perturbation. Autrement dit, on va mesurer la robustesse de cet ordonnancement.

L'étude menée par différents scénarios de transformations de l'ordonnancement initial est présentée pour détecter les faiblesses de l'ordonnancement initial avant sa mise en œuvre.

Lors de la génération des échantillons d'ordonnancement, sont prises en considération les mesures suivantes : la durée de l'ordonnancement (C_{max}), la somme des temps d'attente du robot, la charge totale des bords et la somme des écarts entre le temps de séjour effectif et la

marge minimale.

- **Mesure 1 : Temps de cycle TC**

C'est un critère qui est généralement à minimiser.

- **Mesure 2 : Ecart entre le temps effectif de séjour et la marge minimale**

Il s'agit de l'écart entre le temps de séjour effectif et la marge minimale, défini par :

$$M_{ji}^k = \alpha_{ji}^k - a_{ji}^k \text{ pour } i = 1, \dots, M_j, j = 1, \dots, J, k = 1, \dots, N$$

avec :

a_{ji}^k : borne minimale de traitement pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve ;

α_{ji}^k : temps de séjour effectif pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve ;

La somme des ces écarts est donnée par :

$$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$$

- **Mesure 3 : Temps d'attente total de robot**

Ce critère à minimiser est donné par :

$$W_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$$

δ_{ji}^k étant le temps d'attente du chariot pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit au niveau de la $k^{\text{ème}}$ cuve.

- **Mesure 4 : Charge totale des bains**

Il s'agit de la charge totale des cuves définie par :

$$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$$

Z_{ji}^k étant la charge de la cuve pour la $i^{\text{ème}}$ opération du $j^{\text{ème}}$ produit dans la $k^{\text{ème}}$ cuve.

III.4. Exemple de mise en œuvre : Approche par scénarii

L'exemple suivant expose l'approche par scénarii se basant sur le calcul des mesures de robustesse précisées précédemment.

L'approche par scénarii est une méthode de modélisation des imprévus consistant à considérer un jeu de données pour un problème d'ordonnancement, avec comme objectif la détermination d'une ou de plusieurs solutions qui garantissent des performances sur l'ensemble des jeux des données. Un scénario contient un jeu de données numériques associées à des variables du modèle que l'on suppose susceptible de changer [Rossi, 03].

En plus de la définition du contenu des scénarii, la modélisation des imprévus nécessite la construction d'un ensemble de scénarii noté S_x , x étant le numéro du scénario, chaque scénario est caractérisé par son temps du cycle. S_{xy} reflète l'étendue des données sur lesquelles une garantie de performance de la solution est souhaitée par l'utilisateur pour le scénario S_x .

Il s'agit de proposer 4 scénarii d'ordonnancement S_1, \dots, S_4 , d'injecter une dégradation de vitesse comme élément perturbateur pour l'étude de la robustesse de ces ordonnancements et en distinguer le plus robuste.

Pour chaque scénario S_x , existe alors des sous-scénarii S_{xy} dépendant de l'influence de l'élément perturbateur sur le scénario initial.

Rappelons qu'une fois l'ordonnancement obtenu, les temps des séjours sont fixés avec la vitesse initiale de chariot. Après l'injection de la perturbation consistant à ralentir le chariot, les temps d'attente et les temps de séjours changent et peuvent atteindre les marges maximales. A ce stade l'ordonnancement peut perdre sa robustesse et devient non faisable.

En deuxième phase, une perturbation au niveau des fenêtres du temps de traitement consistant à

dégrader les marges maximales est introduite ; ainsi de nouvelles bornes maximales inférieures à celles adoptées précédemment sont ainsi introduites dès la phase initiale, le but étant d'étudier l'influence de cette dégradation sur la robustesse des ordonnancements.

Au niveau applicatif de l'algorithme développé au chapitre II, à chaque fois qu'un type de perturbation est intégrée, sont recalculés les temps élémentaires de déplacement avec la vitesse dégradée, et est lancé de nouveau le calcul des temps d'attente tout en conservant les données initiales de chaque scénario : le séquençement des opérations et des cuves relatives. Un ordonnancement est considéré faisable si tous les temps d'attente au niveau de toutes les opérations sont positifs.

III.4.1. 1^{ère} perturbation : dégradation de la vitesse

III.4.1.1. Scénario1 : TC= 6785,2s

Le scénario 1, noté S_1 , présente un ordonnancement ayant un temps de cycle égal à 6785,2s défini par une séquence d'opérations et une séquence de cuves. Soit S_{11} ce même ordonnancement avec la vitesse de départ égale à 0,36 m/s et S_{12} avec une vitesse dégradée égale à 0,34 m/s.

Les tableaux III-1 et III-2 suivants présentent les séquençements des opérations et des cuves relatifs au scénario S_{11}

Tableau III-1-Séquençement des opérations relatif au scénario S_{11}

Columns 1 through 18																	
12	27	310	31	32	33	34	35	36	37	38	39	13	14	15	16	17	28
Columns 19 through 36																	
29	210	211	21	22	23	24	25	26	18	19	110	111	112	113	114	115	116
Column 37																	
11																	

Tableau III-2-Séquencement des cuves relatif au scénario S_{11}

Columns 1 through 18																	
12	21	23	1	4	6	22	8	22	13	15	2	3	4	5	6	15	10
Columns 19 through 36																	
11	7	23	1	22	5	6	7	8	14	7	22	18	19	20	7	4	23
Column 37																	
1																	

Les tableaux III-3 et III-4 présentent les résultats de calcul des mesures de robustesses relatives au scénario S_{11} .

Tableau III-3-Valeurs des temps d'attente δ_{ji}^k relatifs au scénario S_{11}

Columns 1 through 11											
0	0	0	0.0600	0.3000	0.0600	0.0600	0.0600	0.0600	0.0600	0.0600	0.1800
Columns 12 through 22											
0.0600	0	0.3000	0.0600	0.0600	0.0600	0	1.8000	0.0600	0.0600	0.0600	0.0600
Columns 23 through 33											
0.3000	0.0600	0.0600	0.0600	0.0600	0.0163	1.8000	0.0600	0.0600	0.0600	0.0600	0.0600
Columns 34 through 37											
0.0600	0.0600	0.0600	0.0600								

Tableau III-4-Valeurs des M_{ji}^k relatifs au scénario S_{11}

Columns 1 through 11											
1.4849	1.6782	0.0518	0	0.0003	0.0382	0.5552	0.1164	0.5552	0	0	
Columns 12 through 22											
0.4639	0	0.0003	0	0.0382	0	0	0.0418	0	0.0518	0	
Columns 23 through 33											
0.5552	0	0.0382	0	0.1164	0	0	0.5552	0	0	0	
Columns 34 through 37											
0	0.0003	0.0518	0								

Le tableau III-5 indique les résultats de calcul des différentes mesures de robustesse Z_{tot} , W_{tot} et M_{tot} à chaque dégradation de vitesse de l'ordonnancement S_1 ayant un temps de cycle de 6785,2.

Tableau III-5-Calcul des mesures de robustesse relatif au scénario S_1

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{11}	0,36	6785,2	6976,5	6196,3	196,5
S_{12}	0,34	6817,3	6995,1	6193,7	215,1
S_{13}	0,32	6853,3	7016,1	6190,9	236,1
S_{14}	0,30	6894,2	7039,8	6187,6	259,8
S_{15}	0,28	6941,0	7066,9	6183,8	286,9
S_{16}	0,26	6995,4	7098,2	6180	318,7
S_{17}	0,24	7063,3	7140,5	6180	360,3
S_{18}	0,22	7143,7	7185,6	6180	409,4
S_{19}	0,20	7240,0	7248,4	6180	468,3
$S_{1,10}$	0,18	7357,8	7320,4	6180	540,4
$S_{1,11}$	0,16	7505,0	7411,0	6180	630,5
$S_{1,12}$	0,14	7694,3	7526,3	6180	746,2
$S_{1,13}$	0,12	7946,7	7680,6	6180	900,6
$S_{1,14}$	0,10	8300,0	7896,7	6180	1116,7
$S_{1,15}$	0,08	8830,0	8221,1	6180	1440,9
$S_{1,16}$	0,06	9713,4	8761,2	6180	1981,2
$S_{1,17}$	0,04	11480,0	11980,8	6180	3061,8
$S_{1,18}$	0,03	0,0	0,0	0,0	0,0
$S_{1,19}$	0,02	0,0	0,0	0,0	0,0

A la vitesse 0,24m/s le scénario S_{17} est toujours faisable avec des temps d'attente positifs.

Tableau III-6-Temps d'attente relatif au scénario $S_{1,17}$

```

tps_cycle =
7.0633e+003

tps_attente_robot =
Columns 1 through 11
      0      0      0      60      300      60      60      60      60      60
Columns 12 through 22
      60      0      300      60      60      60      0      1800      60      60
Columns 23 through 33
      300      60      60      60      60      0      1800      60      60      60
Columns 34 through 37
      60      60      60      60
    
```

A la vitesse 0,03m/s, le scénario $S_{1,18}$ n'est plus faisable avec des temps d'attente négatifs indiqués dans le tableau III-9.

D'après la figure III-1, chaque dégradation de la vitesse entraîne une augmentation de M_{tot} et bien évidemment une augmentation du temps de cycle. Quand la vitesse atteint 0,03m/s, l'ordonnancement n'est plus faisable avec l'apparition d'un temps d'attente négatif, tableau III-7.

Tableau III-7-Temps d'attente relatif au scénario $S_{1,18}$

```

variation_tps_cycle =
0

var_tps_attente_robot =
1.0e+003 *
Columns 1 through 13
      0      0      0      0.0600      0.3000      0.0600      0.0600      0.0600      0.0600      0.0600      0.1800      0.0600      -1.9998
Columns 14 through 26
      0.3000      0.0600      0.0600      0.0600      0      1.8000      0.0600      0.0600      0.0600      0.3000      0.0600      0.0600      0.0600
Columns 27 through 37
      0.0600      0      1.8000      0.0600      0.0600      0.0600      0.0600      0.0600      0.0600      0.0600      0.0600
    
```

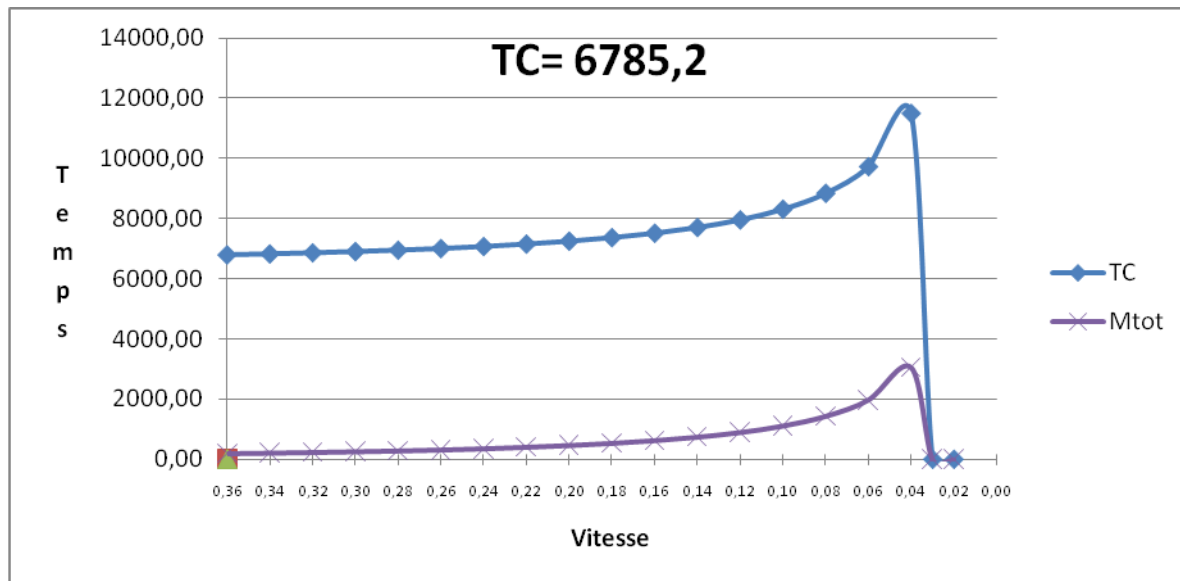


Figure III-1-Mesures de robustesse relatives au scénario S_1

III.4.1.2. Scénario 2 : $TC=6609s$

Un ordonnancement ayant un temps de cycle égal à 6609s correspond au scénario 2, S_2 . Le tableau III-8 suivant indique les résultats de calcul des différentes mesures de robustesse à chaque dégradation de vitesse.

Tableau III-8-Calcul des mesures de robustesse relatif au scénario S_2

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{21}	0,36	6609,0	7735,9	5918,4	955,9
S_{22}	0,34	6643,4	7810,9	5911,2	1030,0
S_{23}	0,32	6686,5	7900,7	5908,6	1120,0
S_{24}	0,30	6736,3	8003,4	5906,5	1223,0
S_{25}	0,28	6793,1	8121,3	5904	1340,0
S_{26}	0,26	6858,8	8254,1	5901,4	1476,0
S_{27}	0,24	6935,3	8419,9	5898,1	1634,0
S_{28}	0,22	0,0	0,0	0,0	0,0
S_{29}	0,20	0,0	0,0	0,0	0,0
$S_{2,10}$	0,18	0,0	0,0	0,0	0,0
$S_{2,11}$	0,16	0,0	0,0	0,0	0,0
$S_{2,12}$	0,14	0,0	0,0	0,0	0,0
$S_{2,13}$	0,12	0,0	0,0	0,0	0,0
$S_{2,14}$	0,10	0,0	0,0	0,0	0,0
$S_{2,15}$	0,08	0,0	0,0	0,0	0,0
$S_{2,16}$	0,06	0,0	0,0	0,0	0,0
$S_{2,17}$	0,04	0,0	0,0	0,0	0,0
$S_{2,18}$	0,03	0,0	0,0	0,0	0,0
$S_{2,19}$	0,02	0,0	0,0	0,0	0,0

La figure III-2 montre une augmentation de M_{tot} et une augmentation du temps de cycle à chaque dégradation de la vitesse. L'algorithme adopté génère un ordonnancement non faisable, correspondant au scénario S_{28} , quand la vitesse atteint 0,08.

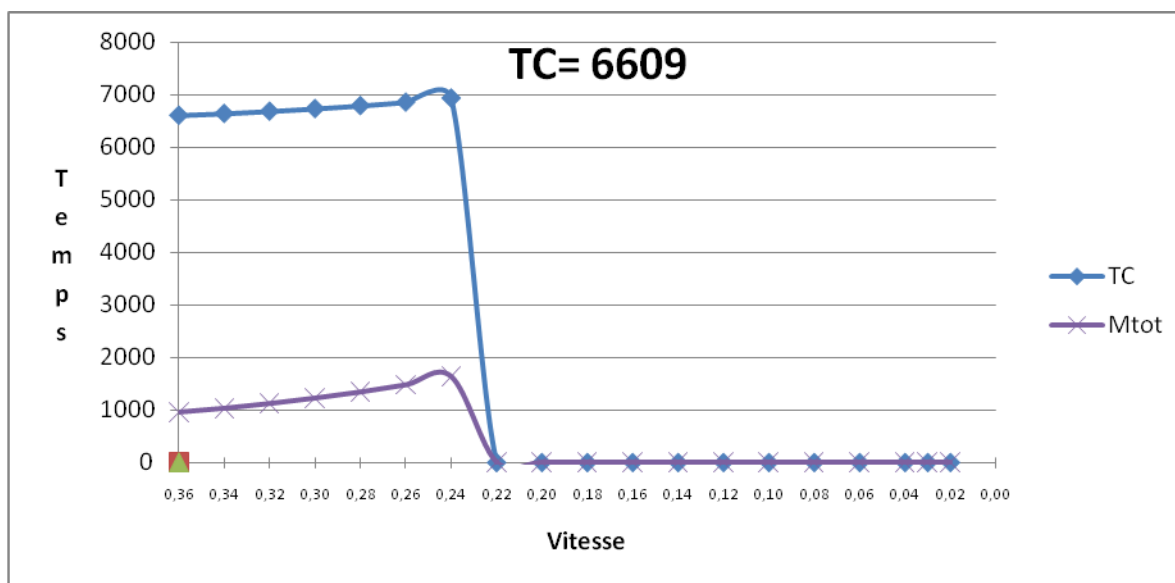


Figure III-2-Mesures de robustesse relatives au scénario S_2

III.4.1.3. Scénario 3 : $TC=6564,9s$

Un ordonnancement ayant un temps de cycle égal à 6564,9s présente le scénario 3, S_3 . Le tableau III-9 suivant indique les résultats de calcul des différentes mesures de robustesse pour chaque dégradation de vitesse.

Tableau III-9-Calcul des mesures de robustesse relatif au scénario S_3

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{31}	0,36	6564,9	7188,3	5989,5	408,3
S_{32}	0,34	6591,0	7219,3	5981,8	439,3
S_{33}	0,32	6620,5	7254,3	5973,2	474,3
S_{34}	0,30	6655,4	7295,5	5965	515,5
S_{35}	0,28	6697,9	7345,4	5958,1	565,1
S_{36}	0,26	6747,0	7401,5	5950	622,5
S_{37}	0,24	6804,2	7469,6	5941,1	689,3
S_{38}	0,22	6871,9	7548,6	5930,4	768,4
S_{39}	0,20	6953,1	7643,2	5917,4	863,2
$S_{3,10}$	0,18	7052,3	7759,1	5901,6	979,0
$S_{3,11}$	0,16	7176,6	7905,1	5882	1124,3
$S_{3,12}$	0,14	7347,7	8102,2	5868,2	1322,2
$S_{3,13}$	0,12	7582,3	8372,5	5856,3	1592,5
$S_{3,14}$	0,10	7910,8	8751,1	5839,6	1971,1
$S_{3,15}$	0,08	8409,1	9324,7	5820	2544,0
$S_{3,16}$	0,06	9272,1	10313	5820	3532,5
$S_{3,17}$	0,04	0,0	0,0	0,0	0,0
$S_{3,18}$	0,03	0,0	0,0	0,0	0,0
$S_{3,19}$	0,02	0,0	0,0	0,0	0,0

La figure III-3 montre que l'ordonnancement du scénario $S_{3,17}$ n'est plus faisable quand la vitesse atteint 0,04. Des augmentations du temps de cycle et du M_{tot} sont remarquées à chaque dégradation de la vitesse.

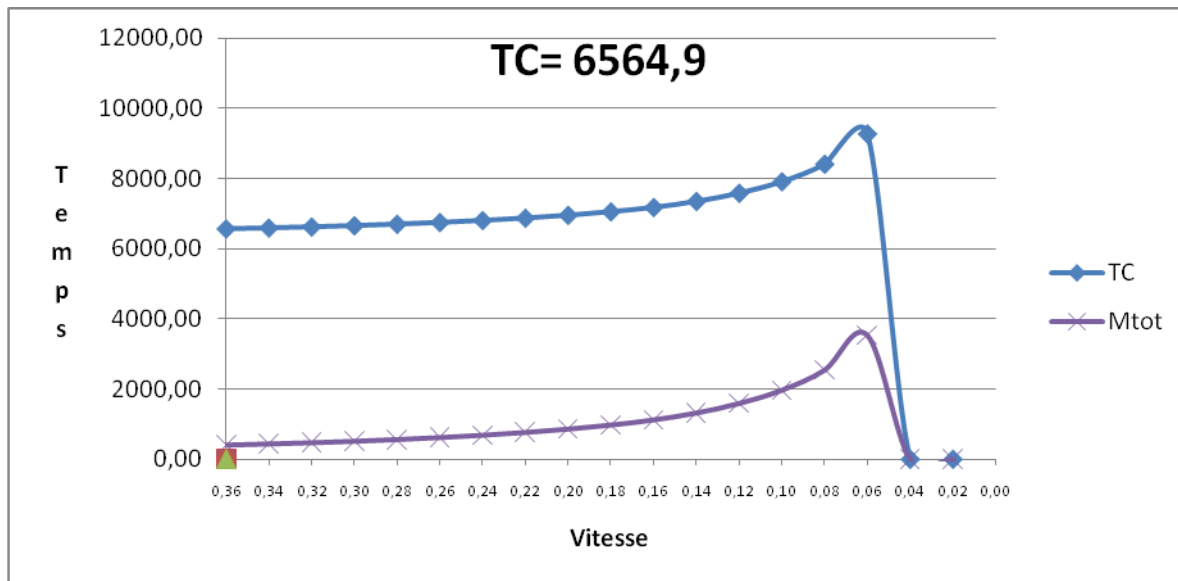


Figure III-3-Mesures de robustesse relatives au scénario S_3

III.4.1.4. Scénario 4 : $TC=7098,1s$

Le scénario S_4 présente un ordonnancement ayant un temps de cycle égal à 7098,1s. Le tableau III-10 suivant indique les résultats de calcul des différentes mesures de robustesse pour chaque dégradation de vitesse.

Tableau III-10-Calcul des mesures de robustesse relatif au scénario S_4

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{41}	0,36	7098,1	7353,4	6427,5	573,4
S_{42}	0,34	7134,4	7394,2	6424,4	614,2
S_{43}	0,32	7175,3	7440,1	6420,9	660,1
S_{44}	0,30	7221,1	7492,1	6417,0	712,1
S_{45}	0,28	7274,6	7551,9	6412,1	771,5
S_{46}	0,26	7335,8	7619,0	6407,4	840,1
S_{47}	0,24	7407,1	7700,5	6401,2	920,1
S_{48}	0,22	7491,4	7795,0	6394,1	1014,7
S_{49}	0,20	7592,5	7908,2	6385,5	1128,2
$S_{4,10}$	0,18	7716,1	8046,9	6375	1266,9
$S_{4,11}$	0,16	7873,5	8224,2	6364	1443,2
$S_{4,12}$	0,14	8084,3	8458,2	6360,0	1678,2
$S_{4,13}$	0,12	8371,6	8777,8	6360,0	1997,8
$S_{4,14}$	0,10	8774,2	9221,4	6360,0	2445,4
$S_{4,15}$	0,09	9042,2	9523,8	6360,0	2743,8
$S_{4,16}$	0,08	0,0	0,0	0,0	0,0
$S_{4,17}$	0,06	0,0	0,0	0,0	0,0
$S_{4,18}$	0,04	0,0	0,0	0,0	0,0
$S_{4,19}$	0,02	0,0	0,0	0,0	0,0

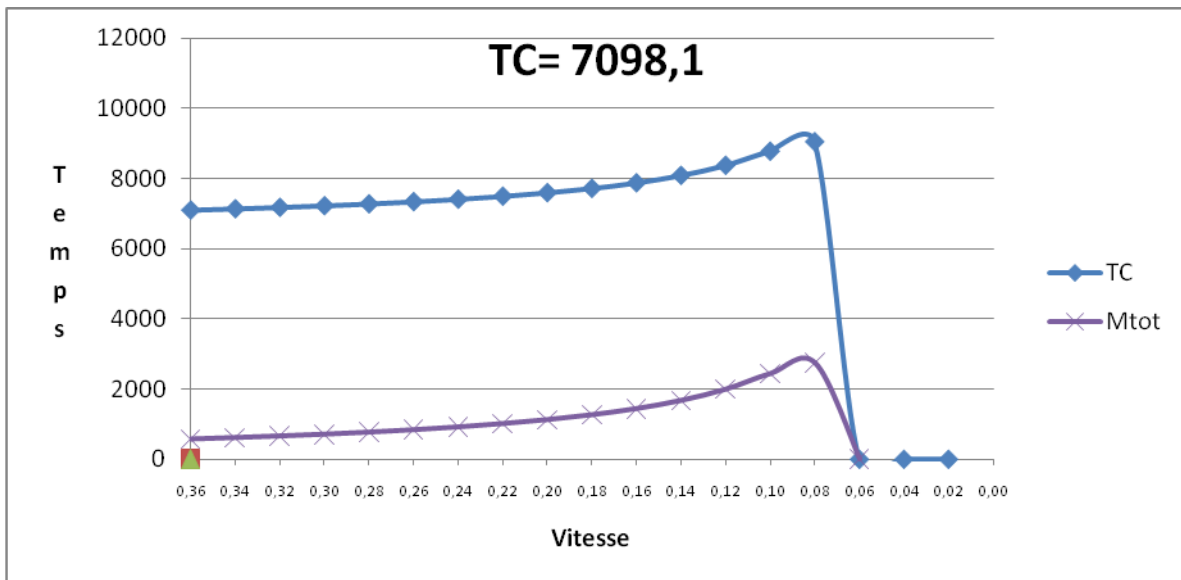


Figure III-4-Mesures de robustesse relatives au scénario S_4

D'après la figure III-4, chaque dégradation de la vitesse entraîne une augmentation de M_{tot} et bien évidemment une augmentation du temps de cycle. Quand la vitesse atteint 0,08m/s, l'ordonnancement relatif au scénario $S_{4,16}$ n'est plus faisable et prend par conséquent une valeur nulle.

En se basant sur la définition de la robustesse qui signifie la résistance au plus d'un ordonnancement face aux perturbations et en se référant alors aux résultats obtenus suite à l'application de l'approche par scénarii, une classification des robustesses des quatre ordonnancements relatifs aux scénarii décrits précédemment est en effet valide. Il vient que S_{11} est celui qui tient le mieux face aux perturbations puisqu'il ne perd sa robustesse qu'à une vitesse de décrochage égale à 0,03 ; il est par conséquent, le plus robuste. Il est suivi de S_{31} qui perd sa robustesse à une vitesse de décrochage égale à 0,05 puis S_{41} qui perd sa robustesse à une vitesse de décrochage égale à 0,08 et enfin S_{21} qui perd sa robustesse à une vitesse de décrochage égale à 0,23.

Le but de la section suivante est de retrouver les résultats précédents mais en se basant sur des indicateurs de performances et sans avoir recours au scénario de ralentissement de la vitesse de chariot pour déterminer la robustesse d'un ordonnancement.

III.4.2. 2^{ème} perturbation : diminution des marges maximales

Soit la nouvelle marge maximale, relative à toutes les opérations, définie par :

$$\left(\text{marge maximale} - \text{marge minimale} / 2 \right) + \text{marge minimale}$$

L'étude de l'influence de cette diminution revient à reconsidérer les scénarii précédents mais en tenant compte des nouvelles marges maximales.

III.4.2.1. Scénario 1 : TC= 6785,2s

Tableau III-11-Calcul des mesures de robustesse relatif au scénario S_1

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{11}	0,36	6785,2	6976,5	6196,3	196,5
S_{12}	0,34	6817,3	6995,1	6193,7	215,1
S_{13}	0,32	6853,3	7016,1	6190,9	236,1
S_{14}	0,30	6894,2	7039,8	6187,6	259,8
S_{15}	0,28	6941,0	7066,9	6183,8	286,9
S_{16}	0,26	6995,4	7098,2	6180	318,7
S_{17}	0,24	7063,3	7140,5	6180	360,3
S_{18}	0,22	7143,7	7185,6	6180	409,4
S_{19}	0,20	7240,0	7248,4	6180	468,3
$S_{1,10}$	0,18	7357,8	7320,4	6180	540,4
$S_{1,11}$	0,16	7505,0	7411,0	6180	630,5
$S_{1,12}$	0,14	7694,3	7526,3	6180	746,2
$S_{1,13}$	0,12	7946,7	7680,6	6180	900,6
$S_{1,14}$	0,10	8300,0	7896,7	6180	1116,7
$S_{1,15}$	0,08	8830,0	8221,1	6180	1440,9
$S_{1,16}$	0,06	0	0	0	0

III.4.2.2. Scénario 2 : TC=6609s
Tableau III-12-Calcul des mesures de robustesse relatif au scénario S_2

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{21}	0,36	6609,0	7735,9	5918,4	955,9
S_{22}	0,34	0	0	0	0

III.4.2.3. Scénario 3 : $TC=6564,9s$

Tableau III-13-Calcul des mesures de robustesse relatif au scénario S_3

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{31}	0,36	6564,9	7188,3	5989,5	408,3
S_{32}	0,34	6591,0	7219,3	5981,8	439,3
S_{33}	0,32	6620,5	7254,3	5973,2	474,3
S_{34}	0,30	6655,4	7295,5	5965	515,5
S_{35}	0,28	6697,9	7345,4	5958,1	565,1
S_{36}	0,26	6747,0	7401,5	5950	622,5
S_{37}	0,24	6804,2	7469,6	5941,1	689,3
S_{38}	0,22	6871,9	7548,6	5930,4	768,4
S_{39}	0,20	6953,1	7643,2	5917,4	863,2
$S_{3,10}$	0,18	7052,3	7759,1	5901,6	979,0
$S_{3,11}$	0,16	7176,6	7905,1	5882	1124,3
$S_{3,12}$	0,14	7347,7	8102,2	5868,2	1322,2
$S_{3,13}$	0,12	7582,3	8372,5	5856,3	1592,5
$S_{3,14}$	0,10	0	0	0	0

III.4.2.4. Scénario 4 : TC=7098,1s

Tableau III-14-Calcul des mesures de robustesse relatif au scénario S_4

Scénario	V	TC	$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$	$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$	$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$
S_{41}	0,36	7098,1	7353,4	6427,5	573,4
S_{42}	0,34	7134,4	7394,2	6424,4	614,2
S_{43}	0,32	7175,3	7440,1	6420,9	660,1
S_{44}	0,30	7221,1	7492,1	6417,0	712,1
S_{45}	0,28	7274,6	7551,9	6412,1	771,5
S_{46}	0,26	7335,8	7619,0	6407,4	840,1
S_{47}	0,24	7407,1	7700,5	6401,2	920,1
S_{48}	0,22	7491,4	7795,0	6394,1	1014,7
S_{49}	0,20	7592,5	7908,2	6385,5	1128,2
$S_{4,10}$	0,18	7716,1	8046,9	6375	1266,9
$S_{4,11}$	0,16	0	0	0	0

III.4.2.5. Etude comparative

En comparant le tableau III-5 (respectivement III-6, III-7, III-8) avec le tableau III-11 (respectivement III-12, III-13, III-14), il est clair que la diminution des marges maximales limite la robustesse de l'ordonnancement en question. Pour le cas de l'ordonnancement ayant un temps de cycle égal à 6785,2s, la vitesse initiale de décrochage de 0,06 devient égale à 0,03 après la diminution des marges maximales. Cette même perturbation agit d'une façon plus marquée sur le scénario 2 qui perd sa robustesse face à cette diminution des fenêtres de temps de traitement ; la vitesse de décrochage initialement égale à 0,22 est devenue égale à 0,34 pour une vitesse initiale de 0,36.

Un classement de ces ordonnancements par rapport à leur robustesse montre que S_{11} constitue le scénario le moins sensible aux perturbations suivi de S_{31} puis de S_{41} et enfin de S_{21} . Ce résultat confirme celui trouvé dans la section précédente, le même classement pouvant être obtenu avec l'intégration d'une nouvelle perturbation.

III.5. Evaluation des performances des solutions

L'analyse de robustesse vise ici à mettre en évidence et à générer des possibilités d'adaptation et de flexibilité pour surmonter les imprévus qui peuvent survenir lors de la mise en œuvre de l'ordonnancement calculé hors ligne. Il s'agit alors de choisir, parmi un ensemble des solutions, celle qui est la plus robuste, relativement à des indicateurs de performances introduits que nous envisageons d'introduire dans le paragraphe suivant.

III.5.1. Formulation des critères

Les critères considérés sont au nombre de trois.

-Le premier indicateur de performance $f1$ est le taux relatif d'occupation moyen par opération :

$$f1 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} M_{ji}^k}{TC * M}$$

-Le deuxième indicateur de performance $f2$ est le taux relatif d'attente moyen du robot par opération :

$$f2 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} \delta_{ji}^k}{TC * M}$$

-Le troisième indicateur de performance $f3$ est le taux relatif d'utilisation moyen du bain par opération :

$$f3 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} M_{ji}^k}{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} Z_{ji}^k * M}$$

Le tableau III-13 regroupe les indicateurs de performances caractérisant la robustesse des scenarii présentés précédemment pour une vitesse de départ 0,36m/s.

Tableau III-15-Calcul des indicateurs de performance

	Scénario S_{11}	Scénario S_{21}	Scénario S_{31}	Scénario S_{41}
<i>TC</i>	6785,2	6609	6564,9	7098,1
Classement de performance par <i>TC</i>	3	2	1	4
$M_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} M_{ji}^k$	196,5	955,9	408,3	573,1
$W = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} \delta_{ji}^k$	34	6.9	16	12
$Z_{tot} = \sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} Z_{ji}^k$	37034	36274	36822	36657
$f1 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} M_{ji}^k}{TC * M}$	1,06	5,35	2,3	2,98
$f2 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} \delta_{ji}^k}{TC * M}$	0,913	0,895	0,912	0,9
$f3 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} M_{ji}^k}{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{Mj} Z_{ji}^k * M}$	1,03	4,57	2,10	2,88
Vitesse de décrochage	0,03	0,23	0,05	0,08

Pour une fonction objectif qui correspond au temps de cycle, le classement des différents scénarii est le suivant S_{31} , S_{21} , S_{11} et finalement S_{41} (ligne 3, tableau III-13).

Dès lors que l'on considère ces mêmes scénarii pour en choisir le plus robuste, il faut s'attendre à ce que chaque scénario conduise à un sous-scénario non faisable ($TC=0$), relatif à la perte de sa robustesse face à l'augmentation de la dégradation de la vitesse du chariot (tableaux III-5, III-8, III-9 et III-10).

Rappelons que, d'après l'approche par scénarii établie dans la section précédente, S_{11} perd sa robustesse (successivement S_{21} , S_{31} , S_{41}) à la vitesse de décrochage 0,03 (successivement 0,23, 0,05, 0,08). Il est clair alors que S_{11} est le plus robuste suivi de S_{31} puis de S_{41} et enfin de S_{21} .

Ce calcul des indicateurs de robustesse vise à chercher lequel parmi ces indicateurs peut déterminer une solution robuste et à fournir un classement mais sans passer par l'approche scénarii.

En fait, la distinction entre différents les scénarii, basée sur un seul critère, ne peut être satisfaite pour en déterminer la meilleure solution.

La nécessité de fournir des solutions, offrant de bons compromis entre les objectifs souvent contradictoires, nous a amené à appliquer l'une des méthodes d'évaluation multicritères, l'intégrale de Choquet, qui permet de comparer et de classer les solutions en tenant compte des différents critères.

III.5.2. Approche d'évaluation multi-critères par l'intégrale de Choquet proposée

L'optimisation multi-objectifs des solutions satisfait simultanément divers objectifs. Ce problème n'a généralement pas une solution unique, mais un ensemble de solutions. Toute solution de cet ensemble est optimale dans le sens qu'aucune amélioration ne peut être faite sur une composante sans dégradation d'au moins une autre composante du vecteur [Talbi, 99]. Compte tenu qu'une solution choisie par un décideur peut ne pas être acceptable par un autre, il s'avère utile de prévoir plusieurs alternatives au choix d'une solution optimale [Zitzler et al, 99], [Rezg et al, 06].

L'approche adoptée consiste à générer une variété de solutions optimales diversifiées dans l'espace de recherche de solutions, et à aider le décideur quand il ne peut pas donner une préférence particulière à l'une des fonctions objectif, en utilisant l'agrégation par l'intégrale de Choquet [Mhedhbi, 08^b].

D'une manière générale, les critères considérés n'ont pas forcément la même importance du point de vue du décideur. Ainsi, beaucoup de considérations peuvent être retenues pour tenir compte de toutes ces difficultés de pondérations des critères et de prise de décision [Saad, 07], [Grabisch, 06].

Définition 1 : Soit $N = \{1, \dots, n\}$ un ensemble de critères. Une capacité sur N est une fonction $\mu : 2^N \rightarrow [0, 1]$ vérifiant $\mu(\emptyset) = 0$, $\mu(N) = 1$ et $\mu(A) \leq \mu(B)$ si $A \subseteq B$ (monotonie).

La condition de monotonie provient du fait que l'importance d'un groupe de critères ne peut décroître si on ajoute un critère au groupe.

Définition 2 : Soit μ une capacité sur N , et $f : N \rightarrow IR$ une fonction représentant les scores d'un objet sur les n critères. L'intégrale de Choquet de f par rapport à μ (score global de l'objet) est donné par :

$$C\mu(f) = \sum_{i=1}^n [f(\sigma(i)) - f(\sigma(i-1))] \mu(A_i)$$

avec $A_i = \{\sigma(i), \dots, \sigma(n)\}$, $f(\sigma(0)) = 0$ et σ une permutation sur N telle que :

$$f(\sigma(1)) \leq f(\sigma(2)) \leq \dots \leq f(\sigma(n))$$

En effet, le choix d'une solution robuste parmi un ensemble de 4 scénarii au niveau d'un ensemble de trois critères constitue un exemple typique de l'interaction entre les critères. L'opérateur de comparaison à utiliser doit tenir compte de l'importance de chaque critère pris à part mais aussi de l'interaction entre les critères. Les intégrales floues, en général, et celle de Choquet, en particulier, permettent de représenter ce genre de comportements. Les priorités du système peuvent être traduites en termes de mesures floues :

1. $\mu(3) = 0.6, \mu(2) = \mu(1) = 0.2$
2. $\mu(2; 3) = 0,3 < \mu(3) + \mu(2)$
3. $\mu(1; 3) = \mu(1; 2) = 0.9 > \mu(1) + \mu(2)$

Les valeurs des trois critères ainsi que les scores obtenus par l'intégrale de Choquet sont donnés dans le tableau III-16. A noter que les critères doivent être ordonnés dans un ordre croissant.

Tableau III-16-Valeurs des scores de l'intégrale de Choquet

	S_{11}	S_{21}	S_{31}	S_{41}
$f2$	0,913	0,895	0,912	0,9
$f3$	1,03	4,57	2,1	2,88
$f1$	1,06	5,35	2,3	2,98
$C\mu$	0,874	2,37	1,296	1,46

Supposons que : $a_1 = f2, a_2 = f3, a_3 = f1$, l'exemple suivant illustre l'application de l'intégrale de Choquet sur le scénario S_{11} et ça sera de même pour les autres scénarii.

$$\begin{aligned}
 C\mu(S_{11}) &= \sum_{i=1}^3 (a_i - a_{i-1}) \mu(\{i, \dots, 3\}) \\
 &= (a_1 - a_0) \mu(\{1, 3\}) + (a_2 - a_1) \mu(\{2, 3\}) + (a_3 - a_2) \mu(\{3, 3\}) \\
 &= 0,874
 \end{aligned}$$

Le tableau III-12, indique que le scénario S_{11} a le meilleur score suivi de S_{31} suivi de S_{41} et enfin de S_{21} . Les scores obtenus par l'intégral de Choquet donnent ce même classement confirmant donc la validité de cette évaluation multicritère dans la détermination d'une solution robuste.

Une solution robuste est en effet celle qui résiste aux dégradations de la vitesse. Plusieurs solutions robustes peuvent exister au sens de cette définition.

On peut alors à ce stade parler de la sensibilité ϕ d'une solution. En effet, ce critère mesure

l'écart observé entre la valeur optimale au sens de la performance mesurée par l'indicateur $C\mu$ et une valeur seuil (suffisante pour l'utilisateur).

Si le critère de performance est généralisé et que l'on souhaite garantir une valeur supérieure à une valeur seuil ϕ_{seuil} égale à 1,3, les deux scénarii 3 et 1, parmi l'ensemble de 4 scénarii, ayant pour valeur de $C\mu$ successivement 0,874 et 1,296, seront maintenus pour être mis en œuvre. Il ne s'agit plus alors de déterminer la solution la plus robuste mais celles qui assurent une garantie de performance suffisante pour l'utilisateur.

III.5.3. Algorithme d'évaluation proposé

L'algorithme présenté ci-dessous génère systématiquement une solution robuste se basant sur les indicateurs de performance précédemment définis. Il constitue la finalisation de celui proposé au chapitre II qui assure la génération d'un ensemble d'ordonnancements grâce à l'algorithme de satisfaction de contraintes hybridé avec SPTF développé.

Pour R implémentations, R ordonnancements, chacun constitué d'un ensemble de 37 opérations et 37 cuves, sont générés.

Niveau 6: Sauvegarder les résultats

- Pour chaque itération r , $r = 1, \dots, R$
- Charger le temps du cycle TC pour chaque ordonnancement S^r , $r = 1, \dots, R$, soit le vecteur TC ($1 \times R$),
- Charger les opérations pour chaque ordonnancement S^r , $r = 1, \dots, R$, soit la matrice OP ($R \times 37$)
- Charger les cuves associées aux opérations pour chaque ordonnancement S^r , $r = 1, \dots, R$, soit la matrice CU ($R \times 37$)
- Charger les temps de séjour effectifs associés aux opérations pour chaque ordonnancement S^r , $r = 1, \dots, R$, soit la matrice TSE ($R \times 37$)
- Charger les bornes minimales associées aux opérations pour chaque ordonnancement

$S^r, r = 1, \dots, R$, soit la matrice BM ($R \times 37$)

- Charger les charges des baigns Z_{ji}^k pour chaque ordonnancement $S^r, r = 1, \dots, R$, soit la matrice CB ($R \times 37$)

- Charger les temps d'attente du robot δ_{ji}^k associés aux opérations pour chaque ordonnancement $S^r, r = 1, \dots, R$, soit la matrice TA matrice ($R \times 37$)

Niveau 7: Formulation des indicateurs de performances

-Calculer la mesure 2 pour chaque opération $M_{ji}^k = \alpha_{ji}^k - a_{ji}^k$ pour $i = 1, \dots, M_j, j = 1, \dots, J, k = 1, \dots, N$

-Calculer la somme de « Mesure 2 » $\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k$ pour chaque ordonnancement $S^r, r = 1, \dots, R$, soit la matrice $Mesure2_r$ ($R \times 1$)

-Calculer la somme des temps d'attente du robot $\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k$ pour chaque ordonnancement $S^r, r = 1, \dots, R$, soit la matrice TA_r ($R \times 1$)

-Calculer la somme des charges des baigns $\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k$ pour chaque ordonnancement $S^r, r = 1, \dots, R$, la matrice CB_r ($R \times 1$)

-Extraire les indices correspondants aux valeurs non nulles de la matrice des temps du cycle TC pour les R itérations, posons un vecteur **indices** (1, size indices), Rappelons que le temps du cycle est nul pour un ordonnancement non faisable.

-Calculer $f1 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k}{TC * M}$

-Calculer $f2 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} \delta_{ji}^k}{TC * M}$

$$\text{-Calculer } f_3 = \frac{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} M_{ji}^k}{\sum_{k=1}^N \sum_{j=1}^J \sum_{i=1}^{M_j} Z_{ji}^k * M}$$

Niveau 8: Application de l'Intégrale de Choquet

-Poser une matrice *Rob* (*R*x3), les éléments de chaque ligne correspondent aux valeurs des indicateurs de performance calculés pour chaque ordonnancement.

<i>Pour r = 1 : R</i>	$f_1(1)$	$f_2(1)$	$f_3(1)$
	.	.	.
	.	.	.
<i>Rob (r,:) =</i>	.	.	.
	.	.	.
	.	.	.
	$f_1(R)$	$f_2(R)$	$f_3(R)$
<i>fin Pour</i>			

-Ordonner les éléments de la matrice *Rob* (*R*x3) dans un ordre croissant au niveau des lignes

-Extraire les valeurs correspondantes au vecteur *indices* pour la matrice *Rob*

$f_1(1)$	$f_2(1)$	$f_3(1)$
.	.	.
.	.	.
.	.	.
$f_1(R)$	$f_2(R)$	$f_3(R)$

$$Rob1 = Rob(indices,:) = Rob(r,:) =$$

-Appliquer l'intégrale de Choquet, initialiser un vecteur $f = zeros(1,1)$

Pour $x = 1$: size (Rob1,1)

$$f(x,1) = Rob1(x,1) * 0.9 + (Rob1(x,2) - Rob1(x,1)) * 0.3 + (Rob1(x,3) - Rob1(x,2)) * 0.6$$

fin Pour

-Ordonner la matrice $f(x,1)$ dans un ordre croissant

- Déterminer ϕ_{seuil}

- Sélectionner l'ensemble des solutions ayant un $f(x,1) \leq \phi_{seuil}$

- L'ordonnancement ayant la valeur la plus petite correspond à la solution la plus robuste.

III.5.4. Résultats de mise en oeuvre

Pour R= 100 implementations , la matrice TC (1xR) présentant la variation du temps du cycle est donnée dans la figure III-5 suivante.

Columns 1 through 15	0	9.0674	0	9.0422	8.9966	8.7642	8.3250	8.9538	8.9416	7.0802	0	8.6358	0	0	6.5390
Columns 16 through 30	6.7360	0	8.5704	8.5222	0	6.5188	0	8.8670	0	8.7124	8.8186	0	8.3320	0	9.3000
Columns 31 through 45	0	8.5572	0	8.6964	0	0	0	8.6698	0	0	0	0	9.1804	8.5942	0
Columns 46 through 60	0	9.3224	8.7500	8.3632	0	8.8922	0	8.6912	0	7.2626	8.6398	0	8.7470	8.5948	8.7294
Columns 61 through 75	0	0	0	0	0	0	0	0	0	0	9.3302	8.5056	0	0	0
Columns 76 through 90	9.0920	9.0570	8.5538	0	8.4898	0	0	0	0	0	0	7.0042	0	0	0
Columns 91 through 100	8.5712	0	0	0	8.3590	0	0	0	0	0					

Figure III-5-Variation du temps du cycle pour R implémentations

Le vecteur indices, correspondant aux valeurs non nulles de la matrice TC ($1 \times R$), est donné dans la figure III-6.

```
indices =  
Columns 1 through 13  
    2     4     5     6     7     8     9    10    12    15    16    18    19  
Columns 14 through 26  
   21    23    25    26    28    30    32    34    38    43    44    47    48  
Columns 27 through 39  
   49    51    53    55    56    58    59    60    71    72    76    77    78  
Columns 40 through 43  
   80    87    91    95
```

Figure III-6-Le vecteur indices

Les valeurs des indicateurs de performances correspondantes au vecteur *indices* sont transmises dans une matrice *Rob1* ($size(indices,1),1$).

$Rob1 = Rob(indices,:) =$

0.2992	0.6617	0.5448
0.3044	0.7034	0.4095
0.2964	0.6803	0.3899
0.2617	0.7120	0.3296
0.3308	0.7640	0.4172
0.0775	0.6902	0.1033
0.1738	0.7007	0.2292
0.3545	0.6259	0.3701
0.1344	0.6862	0.1727
0.4111	0.6278	0.3965
0.6240	0.5958	0.6255
0.3431	0.7055	0.4416
0.3216	0.7181	0.5504
0.2921	0.6719	0.2759
0.1982	0.7037	0.2615
0.1517	0.7110	0.1933
0.0926	0.6992	0.1215
0.1720	0.7417	0.2114
0.1027	0.6798	0.1447
0.1898	0.7362	0.2417
0.1306	0.7106	0.1632
0.1947	0.7405	0.2468
0.3769	0.6993	0.5242
0.5732	0.6912	0.7464
0.1962	0.6887	0.2697
0.3376	0.7261	0.4357
0.2159	0.7390	0.2736
0.1652	0.7220	0.3023
0.2580	0.6808	0.4448
0.3381	0.5948	0.3790
0.1850	0.7153	0.2357
0.2026	0.6989	0.2710
0.0232	0.7406	0.0295
0.3393	0.7354	0.4571
0.3969	0.6642	0.5367
0.1204	0.7125	0.1511
0.2565	0.6863	0.3440
0.4176	0.6956	0.5678
0.2139	0.7014	0.3630
0.2885	0.7121	0.3612
0.3118	0.6168	0.3221
0.2876	0.7000	0.4892
0.2386	0.6911	0.2941

L'application de l'intégrale de Choquet génère la matrice $f(size(indices,1)x1)$ qui est directement ordonné dans un ordre croissant.

$$f(x,1) = Rob1(x,1)*0.9 + (Rob1(x,2) - Rob1(x,1))*0.3 + (Rob1(x,3) - Rob1(x,2))*0.6 =$$

```

0.4131
0.4261
0.4296
0.4298
0.4386
0.4403
0.4406
0.4416
0.4458
0.4461
0.4495
0.4500
0.4544
0.4558
0.4559
0.4587
0.4596
0.4597
0.4605
0.4625
0.4627
0.4690
0.4694
0.4696
0.4757
0.4772
0.4811
0.4818
0.4831
0.4848
0.4854
0.4871

0.4885
0.4908
0.4913
0.4920
0.4967
0.4975
0.5075
0.5077
0.5317
0.5456
0.5844

```

$f(1,1)$ correspond au plus petit score obtenu par l'application de l'intégrale de Choquet relatif à la solution la plus robuste. Les données associées à cette solution sont ultérieurement extraites pour la mise en œuvre en ligne.

Les différents résultats obtenus, indiqués dans les figures et tableaux précédents, montrent l'efficacité de l'approche proposée devenant plus intéressante dès lors qu'on a un nombre plus grand d'implémentations puisqu'il génère un ensemble de solutions robustes plus important pour une valeur seuil donnée.

III.6. Conclusion

Pour l'ordonnancement d'un atelier de traitement de surfaces, les contraintes de temps fixent un critère d'admissibilité strict et en présence d'aléas et de perturbations, la mise en œuvre de la robustesse de l'ordonnancement devient utile pour maintenir une production correcte vis-à-vis du

cahier des charges.

Pour ce fait, après les définitions formelles des notions de robustesse et de flexibilité, une approche par scénarii est présentée pour l'étude de la robustesse d'un ordonnancement donné face à des perturbations rencontrées lors de sa mise en œuvre,

Quatre scénarii, générés par l'algorithme de satisfaction de contraintes et ayant chacun un temps de cycle, sont choisis pour illustrer cette approche.

Le chariot étant la ressource critique de cet atelier, la dégradation de sa vitesse est retenue et a permis d'étudier l'effet de cette perturbation sur la robustesse des scénarii, pouvant être le dépassement des marges maximales et donc l'endommagement du produit.

Les résultats de la classification de ces scénarii en fonction de leur résistance à cette perturbation ont été confirmés pour une perturbation relative à une diminution des marges maximales dès la phase initiale.

Des mesures de robustesse, proposées dans ce chapitre, ont permis, par une évaluation multi-critères basée sur l'intégrale de Choquet, de déterminer le scénario le plus robuste parmi un ensemble de solutions faisables.

La sélection systématique de l'ordonnancement robuste a été enfin effectuée grâce à un algorithme élaboré, ne nécessitant pas le recours à l'approche par scénarii.

CONCLUSION GENERALE

L'élaboration d'ordonnements des ateliers de traitement de surfaces consistant à trouver les séquences des produits à traiter et des ressources pour minimiser le temps du cycle tenant compte des spécificités des lignes étudiées dont les contraintes de précédence et les contraintes de ressources, constituent la principale contribution de nos travaux de recherche.

Nos travaux de recherche ont permis de définir les performances des méthodes de résolution de problèmes d'ordonnement de type job shop en particulier des problèmes mono-robot/multi-produits connu sous le nom Hoist Scheduling Problem (HSP), appliquées à une chaîne de traitement de surfaces.

Après la présentation des lignes de traitement de surfaces étudiées, des différents types de production pouvant caractériser ce type d'ateliers ainsi que des spécificités des lignes de galvanoplastie considérées dans nos travaux, les méthodes de modélisation les plus adaptées pour la présentation d'un problème d'ordonnement sont ensuite précisées. Un état de l'art est enfin élaboré sur les méthodes de résolution des problèmes HSP allant des méthodes exactes aux méthodes approchées.

Une modélisation globale est élaborée, dans le deuxième chapitre, à partir des gammes logiques et une modélisation plus explicite partant des gammes opératoires restreintes à partir d'une description détaillée de la ligne de traitement de surfaces étudiée mettant en exergue les ressources du système, les opérations de traitement, les temporisations et les contraintes.

Trois approches sont mises en œuvre pour la détermination d'ordonnements satisfaisants; d'abord un algorithme de satisfaction de contraintes est proposé, exploité dans une deuxième phase par hybridation avec d'autres heuristiques et enfin les algorithmes génétiques.

L'utilisation de l'algorithme de satisfaction de contraintes proposé, constituant une méthode de résolution exacte, s'est avérée intéressante pour les problèmes contraints et efficace pour déterminer une solution admissible lors des ajouts de nouvelles contraintes.

Nous avons montré également que l'hybridation avec d'autres méthodes conduit à de meilleures solutions et que l'efficacité des algorithmes génétiques pour l'optimisation du problème job shop mono-robot/multi-produits étudié est encore plus significative. Des essais numériques et leurs résultats ont validé les approches considérées.

Sont exploitées enfin les notions de robustesse et de stabilité, introduites en présence d'une perturbation au niveau de la ressource critique de l'atelier qui est le chariot.

Pour l'atelier de traitement de surfaces étudié, plusieurs mesures de la robustesse ont été, en effet, considérées en rapport avec la dégradation de la vitesse du chariot considérée comme une perturbation spécifique.

La distinction de divers scénari a été nécessaire pour l'étude de l'influence de cette perturbation sur les mesures retenues de la robustesse.

La détermination systématique d'un ordonnancement robuste a été menée, avec succès, par introduction de nouveaux indicateurs de performances et par application d'une méthode d'évaluation multicritère.

Les résultats obtenus étant encourageants dans le sens de l'obtention de solutions satisfaisantes, il serait intéressant de les développer, en perspectives, dans le sens de :

- la généralisation de l'étude de la robustesse de l'ordonnancement, relative aux perturbations de la dégradation de la vitesse du chariot et à la dégradation des marges maximales, à celle relative à d'autres types de perturbations,
- l'exploitation d'autres métaheuristiques telles que les essais particuliers, le recuit simulé ou la recherche tabou et la comparaison de leurs résultats de mise en œuvre avec ceux, satisfaisants, obtenus par les algorithmes génétiques,
- la comparaison de l'évaluation multicritère basée sur l'intégrale de Choquet, considérée dans nos travaux, avec celles relatives à d'autres méthodes telles que la méthode de pondération, l'agrégation par les opérateurs d'agrégation OWA,
- l'étude des régimes transitoires dans la conception de l'ordonnancement cyclique de l'atelier de traitement de surfaces pour l'optimisation de sa production et sa productivité,
- la mise en œuvre des résultats obtenus et leur adaptation au processus industriel considéré au démarrage de nos travaux de recherche.

BIBLIOGRAPHIE

- [Alcaide et al, 02] Alcaide, D., Rodriguez-Gonzalez, A., Sicilia, J., «An approach to solve the minimum expected makespan flowshop problem subject to breakdowns», *European Journal of Operational Research*, Vol. 140, p. 384-398, 2002.
- [Aloulou, 02] Aloulou, M.A., «Structure flexible d'ordonnements à performances contrôlées pour le pilotage d'atelier en présence de perturbations», Thèse de Doctorat de l'Institut National Polytechnique de Lorraine, Nancy - France, 2002.
- [Amar, 94] Amar S., «Systèmes Automatisés et flexibles de production manufacturière : méthode de conception du système de coordination par prototypage orienté objet de la partie procédé», Thèse de Doctorat en Productique, Automatique et Informatique Industrielle, Université de Lille 1, 1994.
- [Artigues et al, 02] Artigues, C., Briand, C., Portmann, M.C., Roubellat, F., «Pilotage des systèmes de production», chapitre Pilotage d'atelier basé sur un ordonnancement flexible, Edition Hermes, Kieffer J.P. & Pujo P., 2002.
- [Artigues et al, 97] Artigues, C., Roubelat, F., «An operation insertion procedure in a multiresource schedule based on a dominance rules», Tech. rep., LAAS, 01 1997.
- [Baillet, 94] Baillet, P., «Contribution à l'amélioration de la réactivité dans les systèmes de production notamment par la mise en œuvre de concepts de décentralisation des fonctions de décision», Thèse de Doctorat soutenue à l'Université d'Aix Marseille III, France, 1994.
- [Balasubramanian et al, 02] Balasubramanian, J., Grossmann, I.E., «A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing

-
- time», *Computers and Chemical Engineering*, Vol. 26, p. 41-57, 2002.
- [Baptiste et al, 01] Baptiste P., Bloch C., Varnier C., Chapitre 9 : «Ordonnancement des lignes de traitement de surfaces», *Traité IC2, Information-Commande-Communication, section Productique, Ordonnancement de la Production*, coordonné par F. Roubellat et P. Lopez, Ed. Hermès-Systèmes pour l'Ingénieur, p. 259-293 ,2001.
- [Baptiste et al, 96] Baptiste, P., Legeard, B., Manier, M.A., Varnier, C., «Résolution d'un problème d'ordonnancement avec la PLC», *Journal Européen des Systèmes Automatisés* 30, 2-3, p. 201-30,1996.
- [Barecke, 09] Barecke, T., «Isomorphisme inexact de graphes par optimisation évolutionnaire», *Thèse de Doctorat, Université Paris VI*, 2009.
- [Benbouzid, 05] Benbouzid. F., «Contribution à l'étude de la performance et de la robustesse des ordonnancements conjoints production / maintenance - cas du flowshop», *Thèse de Doctorat, Université de Franche-Comte*, 2005.
- [Blazewicz et al, 94] Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J., «Scheduling Computer in Manufacturing Processes», Springer, Berlin, 1994.
- [Bloch, 99] Bloch, C., «Contribution à l'ordonnancement dynamique de lignes de traitement de surface», *PhD Thesis, Université de Franche-Comté*, 1999.
- [Boukef et al, 09] Boukef, H., Borne, P., Benrejeb, M., «Genetic algorithm and based Particle swarm Optimization comparison for solving coding flow-shop multi objective scheduling problems in pharmaceuticals industries», *International Review of Automatic Control*, 2(2), p. 223-228, 2009.
- [Brailsford et al, 99] Brailsford, S.C., Potts, C.N., Smith, B.M., «Constraint satisfaction problems: Algorithms and applications», *European Journal of Operational Research*, 119, p. 557-581, 1999.
- [Branke et al, 00] Branke , J., Mattfeld, D.C., «Anticipation in dynamic optimization: The scheduling case», *Proceeding of sixth Parallel Problem Solving*

-
- from Nature, LNCS Vol. 1917, p. 253-262, Springer, 2000.
- [Brauner et al, 97] Brauner, N., Finke, G., Kubiak, W., «A proof of the Lei and Wang claim», Tech. rep., Laboratoire Leibniz, Institut IMAG, 1997.
- [Bruns, 93] Bruns, R., «Direct chromosome representation and advanced genetic operators for production scheduling», 5th International Conference on Genetic Algorithms, University of Illinois at Urbana Champaign, p. 352-359, 1993.
- [Camus, 97] Camus, H., «Conduite de systèmes flexibles de production manufacturière par composition de régimes permanents cycliques : Modélisation et évaluation de performances à l'aide des Réseaux de Petri», Thèse de Doctorat, Université des Sciences et Technologies de Lille, 1997.
- [Carlier et al, 84] Carlier, J., Chrétienne, P., Girault, C., «Modelling scheduling problems with Petri nets», Advances studies in Petri nets. Lecture notes in Computer Science, Springer Verlag, Paris, 1984.
- [Carlier et al, 88] Carlier, J., Chrétienne, P., «Problème d'ordonnancement : Modélisation, Complexité, Algorithmes», Edition Masson, Paris, 1988.
- [Charon et al, 96] Charon I., A. Germa, O. Hudry, «Méthodes d'optimisation combinatoire », Éditions Masson, Paris, 1996.
- [Chauvet et al, 00] Chauvet, F., Levner, E., Meyzin, L. K., Proth, J.M., «On-line scheduling in a surface treatment system», European Journal of Operational Research 120, p. 382-392, 2000.
- [Che et al, 01] Che, A., Chu, F., Chu, C., «Un algorithme d'ordonnancement multi-cycliques d'un robot», MOSIM , France, 2001.
- [Chen et al, 94] Chen, H., Chu, C., Proth, J.-M., «Cyclic scheduling of a hoist with the time window constraint», Tech. Rep. 2307, INRIA, 1994.
- [Chen et al, 95] Chen, H., Chu, C., Proth, J.M., «Cyclic hoist scheduling based on graph theory», INRIA/IEEE Symposium on Emerging Technologies and Factory Automation, Vol. 1, p. 451-459, 1995.
- [Chen et al, 97] Chen, H., Chu, C., Proth, J.-M., «Sequencing of parts in robotic cells»,

-
- International Journal of Flexible Manufacturing Systems , 1997.
- [Chen et al, 99] Chen, H., Ihlow, J., Lehmann, C., «A genetic algorithm for flexible job shop scheduling», IEEE International Conference on Robotics and Automation, p. 1120-1125, Detroit, USA, 1999.
- [Cheng et al, 94] Cheng ,C., Smith, S.F., «Generating feasible schedules under complex metric constraints», in Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle, Washington, USA, 1994.
- [Cheng et al, 97] Cheng, C., Smith, S.F, «Applying constraint satisfaction techniques to job shop scheduling», Annals of Operations Research 70, p. 327-357, 1997.
- [Collart-Dutilleul et al, 97] Collart-Dutilleul S., Denat J.P., «P-time Petri net and robust control of electroplating lines using several hoists», Proceedings of the IFAC International Conference on Control of Industrial Systems (IFAC-IFIP-IMACS CIS'97), Vol.3, p. 501-506, France, 1997.
- [Collart-Dutilleul, 97] Collart-Dutilleul, S., «Commande robuste d'ateliers à contraintes de temps de séjour : application à la galvanoplastie», Thèse de doctorat, Ecole Supérieure d'Ingénieur d'Annecy, 1997.
- [Collart-Dutilleul, 08] Collart-Dutilleul, S., «Les Réseaux de Petri P-temporels Modélisation et validation des exigences temporelles», HdR, USTL, Lille, 2008.
- [Collette et al, 02] Collette,Y., Siary,P., «Optimisation multiobjectif», Edition Eyrolles, Paris, 2002.
- [Cowling et al, 02] Cowling, P. I., Johansson, M., «Using real time information for effective dynamic scheduling», European Journal of Operational Research, Vol. 139(2), p. 230-244, 2002.
- [Davenport et al, 00] Davenport, A.J., Beck J.C., «A survey of techniques for scheduling with uncertainty»,<http://eil.utoronto.ca/profiles/chris/chris.papers.html>, 2000.
- [Davis, 91] Davis, L.D, «Handbook of Genetic Algorithms», Van Nostrand Reinhold Company, 1991.

-
- [Dechter et Frost,02] Dechter, R., Frost, D., «Backjump-based Backtracking for Constraint Satisfaction Problems», *Artificial Intelligence*, p.147-188, 2002.
- [Fleury et al, 01] Fleury, G., Gourgand, M., Lacomme, P., «Meta-heuristics for the stochastic hoist scheduling problem», *International Journal of Production Research* 39, 15, p. 3419-3457, 2001.
- [Fleury et al, 99] Fleury, G., Gourgand, M., and Lacomme, P., «Coupling meta-heuristics and discrete event simulation models for the stochastic hoist scheduling problem », *Advanced Computer Systems, ASC'99*.
- [Ghedjati, 94] Ghedjati, F., «Résolution par des heuristiques dynamiques et des algorithmes génétiques du problème d'ordonnancement de type job-shop généralisé», Thèse de Doctorat, Université de Paris VI, 1994.
- [Goldberg, 89] Goldberg, D., «Genetic Algorithm in Search», Edition Addison Wesley, France, 1989.
- [Goldberg, 94] Goldberg, G.E, «Algorithmes Génétiques», Edition Addison Wesley, France, 1994.
- [Gotha, 93] GOTHA, «Les problèmes d'ordonnancement», *RAIRO Recherche Opérationnelle*, 27(1), p.77-150, 1993.
- [Goujon et al, 97] Goujon, J.Y., Lacomme, P., «Hoist scheduling problem : Modèle objets», Rapport interne, Laboratoire d'Informatique (LIMOS), 1997.
- [Grabisch, 06] Grabisch, M., Groupe de Travail Européen, «Aide Multicritère à la Décision», Série 3, n°14, 2006.
- [Graham, 69] Graham R.L., «Bounds on multiprocessing timing anomalies», *SIAM journal on Applied Mathematics*, p. 416-429, 1969
- [Grunder et al, 97] Grunder, O., Baptiste, P., Chappe, D., «The relationship between the physical layout of the work stations and the productivity of a saturated single-hoist production line», *International Journal of Production Research*, 35(8), p. 2189-2211, 1997.
- [Herroelen et al, 02] Herroelen, W., Leus, R., «Project scheduling under uncertainty survey and research potentials», *Eighth International Workshop on Project Management and Scheduling, Spain*, 2002.

-
- [Hindi et al, 04] Hindi, K.S., Fleszar, K., «A constraint propagation heuristic for the single-hoist multiple-products scheduling problem», *Computers & Industrial Engineering* 47, p. 91-101, 2004.
- [Holland , 75] Holland, J.H, «Adaptation in Natural and Artificial System», The University of Michigan Press, 1975.
- [Huber, 81] Huber P. J., «Robust Statistics», John Wiley & Sons, 1981
- [Jensen, 00] Jensen, M.T., «Neighbourhood based robustness applied to tardiness and total flowtime jobshops», *Proceeding of sixth Parallel Problem Solving from Nature, LNCS, Vol. 1917*, p. 283-292, Springer, 2000.
- [Jensen, 01] Jensen, M.T., «Improving robustness and flexibility of tardiness and total flowtime jobshop using robustness measures», *Applied Soft Computing, Vol. 1*, p. 35-52, 2001.
- [Jerbi, 06] Jerbi, N., «Apports et intégration de la robustesse pour la supervision de systèmes manufacturiers», Thèse de Doctorat, Ecole Nationale d'Ingénieurs de Tunis, Tunisie , 2006.
- [Jourdan et al, 09] Jourdan, L., Basseur., M., Talbi , E.G., «Hybridizing exact methods and metaheuristics : A taxonomy», *European Journal of Operational Research* 199, p. 620-629, 2009.
- [Kaabi, 04] Kaabi, J., «Contribution à l'ordonnancement des activités de maintenance dans les systèmes de production», Thèse de Doctorat, Université de Franche-Comté, 2004.
- [Kacem, 03] Kacem I., «Ordonnancement multicritère des job-shops flexibles : formulation, bornes inférieures et approche évolutionniste coopérative», Thèse de Doctorat, USTL 1, 2003.
- [Kackar, 85] Kackar R. N., «Off-line Quality control, Parameter design and the Taguchi Method», *Journal of Quality Technology, Vol.17 (4)*, p. 176-188, 1985
- [Kats et al, 99] Kats V., Levner E., Meyzin L., «Multiple-part cyclic hoist scheduling using a sieve method», *IEEE Transactions on Robotics and Automation, 15(4)*, p.704-713, 1999.

-
- [Khansa, 97] Khansa, W., «Réseaux de Petri p-temporels : contribution à l'étude des systèmes à événements discrets», Thèse de Doctorat, Ecole Supérieure d'Ingénieur d'Annecy, 1997.
- [Knosala et al, 01] Knosala, R., Wal, T., «A production scheduling problem using genetic algorithm», Journal of Materials Processing Technology, Vol. 109, p. 90-95, 2001.
- [Kobayashi et al, 95] Kobayashi, S., Ono, I., Yamamura, M., «An efficient genetic algorithm for job-shop scheduling problems», 6th International Conference on Genetic Algorithms, p. 506-511, 1995.
- [Korbaa, 98] Korbaa, O., «Commande Cyclique des Systèmes Flexibles de Production Manufacturière à l'aide des Réseaux de Petri : De la Planification à l'Ordonnancement des Régimes Transitoires», Thèse de Doctorat, Université de Sciences et des Technologies de Lille, 1998.
- [Korbaa, 03] Korbaa, O., «Contribution à la conception et l'optimisation des systèmes de transport et de production», HdR, USTL, Lille, 2003.
- [Kouvelis et al, 00] Kouvelis, P., Daniel, R.L., Vairaktarakis, G., «Robust Scheduling of a two-machine flow shop with uncertain processing times», IIE Transactions, p. 421-432, 2000.
- [Lamothe, 96] Lamothe, J., «Une approche pour l'ordonnancement dynamique d'un atelier de traitement de surfaces», Thèse de Doctorat, Ecole Supérieure de l'Aéronautique et de l'Espace, Toulouse, 1996.
- [Lawrence et al, 97] Lawrence, S.R., Sewell, E.C., «Heuristic, optimal, static and dynamic schedules when processing times are uncertain», Journal of Operations Management, Vol. 15, p. 71-82, 1997.
- [Lei et al, 89] Lei, L., Wang, T.J., «A proof : the cyclic HSP is NP-complete», Tech. Rep. 89-0016, Graduate School of Management, Rutgers University, 1989.
- [Lei et al, 91] Lei, L., Wang, T.J., «The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints», Management Science, 37(12), p.1629-1639, 1996.

-
- [Lei et al,94] Lei, L., Wang, T.J., «Determining optimal cyclic hoist schedules in a single-hoist electroplating line», IIE Transactions, 26/2 , p.25-33, 1994.
- [Lenstra et al, 77] Lenstra, J.K., Rinnooy Kan A.H.G., Brucker P., «Complexity of machine scheduling problems», Annals of Discrete Mathematics, 1, p. 343-362, 1977.
- [Levner et al, 98] Levner E., Kats V., «A parametric critical path problem and an application for cyclic scheduling», Discrete Applied Mathematics, 87, p.149-158, 1998.
- [Lim , 97] Lim, J.M., «A genetic algorithm for a single hoist scheduling in the printed-circuit- board electroplating line», Computers & Industrial Engineering ,Vol. 33, N° 3-4, p.789-792, 1997.
- [Lopez et al, 00] Lopez, P., Roubelat, F, «Ordonnement de la production», Lavoisier, Hermès, 2000.
- [Lopez et al, 99] Lopez, P., Esquirol, P., «L'ordonnement», Edition Economica, Paris, 1999.
- [Mangione, 03] Magnione, F., «Ordonnement des ateliers de traitement de surface pour une production cyclique et mono-produit», Thèse de Doctorat, Institut National Polytechnique de Grenoble, 2003.
- [Manier et al, 94] Manier, M.A., Baptiste, P., «Etat de l'art : Ordonnement de robots de manutention en galvanoplastie», APII 28, 1, p.7-35, 1994.
- [Mehta et al, 99] Mehta, S.V., Uzsoy, R., «Predict scheduling of a single machine subjects to breakdowns», International Journal of Computer Integrated Manufacturing, Vol. 12, p. 15-28, 1999.
- [Mellouli et al, 04] Mellouli, K., El Kamel, A., Borne, P., « Programmation linéaire et applications», Eléments de cours et exercices. Edition Technip, Paris, 2004.
- [Mesghouni, 99] Mesghouni, K., «Application des algorithmes évolutionnistes dans les problèmes d'optimisation en ordonnancement de la production», Thèse de Doctorat, Université des Sciences et Technologies de Lille 1,

-
- 1999.
- [Mhedhbi et al, 08^a] Mhedhbi, I., Camus, H., Craye, E., Benrejeb, M., «On SHMP scheduling of surfaces treatment line based on margins measures», IJ-STA, Vol. 2, N° 2, p. 696-709, 2008.
- [Mhedhbi et al, 08^b] Mhedhbi, I., Camus, H., Craye, E., Benrejeb, M., « Evaluation multicritère d'ordonnancement d'un atelier de traitement de surfaces en utilisant l'intégrale de Choquet », SENDA 08, Monastir, 2008.
- [Mhedhbi et al, 10] Mhedhbi, I., Camus, H., Craye, E., Benrejeb, M., « Solving Job Shop Scheduling Problem in a surface treatment line using an hybrid constraints satisfaction algorithm», CCCA'11, Hammamet, 2011.
- [Mhedhbi et al, 11] Mhedhbi, I., Camus, H., Craye, E., Benrejeb, M., «Proposed constraints satisfaction algorithm for solving job shop scheduling problem in a surface treatment line», accepté à International Journal of Production Research, 2011.
- [Mhedhbi et al, 11^a] Mhedhbi, I., Camus, H., Craye, E., Benrejeb, M., « SHMP problem solution for surface treatment line using genetic algorithm», International Review of Automatic Control, IREACO, 2011.
- [Mignon et al, 95] Mignon, D.J., Honkomp, S.J., Reklaitis, G.V., «A framework for investigating schedule robustness under uncertainty», Computers and Chemical Engineering, Vol.21, Suppl., p. S615-S620, 1995.
- [Minton et al, 92] Minton, S., Johnston, M.D., Philips, A.B., Laird, P., «Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems», Artificial Intelligence, 58, p.161-205, 1992.
- [Murata, 89] Murata T., «Petri nets_ Properties, analysis, and applications», Proceedings of the IEEE, Vol 77 N°4, p.541-580, 1989.
- [NG, 96] NG, W., «A branch and bound algorithm for hoist scheduling of a circuit board production line», The International Journal of Flexible Manufacturing Systems, Vol. 8, p. 45-65, 1996.
- [Ohl, 95] Ohl, H., «Fonctionnements répétitifs de systèmes flexibles de production manufacturière : Analyse et Optimisation des performances

-
- à l'aide des Réseaux de Petri», Thèse de Doctorat, Université des Sciences et Technologies de Lille 1, 1995.
- [Pezzella et al, 08] Pezzella, F., Morganti, G., Ciaschetti, G., «A genetic algorithm for the flexible job-shop scheduling problem», *Computers & Operations Research* 35, p. 3202-3212, 2008.
- [Phillips et al, 76] Phillips, L., Hunger, P., «Mathematical programming solution of a hoist scheduling problem», *AIIE Transactions*, p. 219–225, 1976.
- [Pinedo, 05] Pinedo, M., «Planning and scheduling in manufacturing and services», Springer, 2005.
- [Pinedo, 95] Pinedo, M., «Scheduling Theory, Algorithms and Systems», Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [Pistikopoulos et al, 96] Pistikopoulos, E.N., Thomaidis, T.V., Melin, A. Ierapetritov, M.G., «Flexibility, reliability and maintenance considerations in batch plant design under uncertainty», *Computers and Chemical Engineering*, Vol. 20, Suppl., p. 1209-1214, 1996
- [Pomerol, 01]. Pomerol, J.C., «Scenario development and practical decision making under uncertainty», *Decision Support Systems*, Vol. 31, p. 197-204, 2001
- [Portmann, 88] Portmann, M.C., «Méthodes de décompositions spatiale et temporelle en ordonnancement de la production», *APII*, Vol. 22, 1988.
- [Pujo et al, 02] Pujo, P., Kieffer, J.P. «Méthodes de pilotage des systèmes de production», Lavoisier, Hermes, 2002.
- [Rezg et al, 06]. Rezg, N., Boschian, V. et Chelbi A., «Evaluation des performances d'un système manufacturier : approche multicritères», *Journal européen des systèmes automatisés*, Vol 40, N°7, p. 787-809, 2006,
- [Rodammer et al, 88] Rodammer, F. A., White, K. P., « A recent survey of production scheduling», *IEEE Transactions on Systems, Man, and Cybernetics*, p. 6-18, 1988.
- [Ross, 97] Ross, P., «What are genetic algorithms good», *Inform Journal on Computing*, Vol. 9, N °3, 1997.

-
- [Rossi, 03] Rossi, A., « Ordonnancement en milieu incertain, mise en oeuvre d'une démarche robuste », Thèse de Doctorat, Grenoble, 2003.
- [Roy et al, 93] Roy, B., Bouyssou, D., « Aide Multi-critères à la décision : Méthodes et cas », Collection Gestion Série : Production et technologie quantitative appliquées à la gestion. Edition Economica, Paris, 1993.
- [Roy, 70] Roy, B., « Algèbre moderne et théorie des graphes », Vol 2. Dunod, Paris, 1970.
- [Roy, 97] Roy, B., « Un chaînon manquant en RO-AD : les conclusions robustes », Cahier du Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la Décision, N° 144, Université de Paris Dauphine, 1997.
- [Saad, 07] Saad, I., « Conception d'un système d'aide à l'ordonnancement tenant compte des impératifs économiques », Thèse de Doctorat, Ecole Centrale de Lille et Ecole Nationale d'Ingénieurs de Tunis, 2007.
- [Shafaei et al, 99a], Shafaei, R., Brun, P., « Workshop scheduling practical (inaccurate) data. Part 1: An investigation of robustness of scheduling rules in dynamic jobshop environment using a rolling time horizon approach », International Journal of Production Research, Vol. 37(17), p. 3913-3925, 1999.
- [Shapiro, 85] Shapiro, G.W., « Hoist scheduling for a PCB Electroplating Facility », Program in operation research, Graduate Faculty of North Carolina State University, 1985.
- [Shapiro, 88] Shapiro, G.W., Nuttle, H., « Hoist scheduling for a pcb electroplating facility », IIE Transactions 20, 2, p.157-167, 1988.
- [Smith et al, 93] Smith, S.F., Cheng, C., « Slack-based heuristics for constraint satisfaction scheduling », in Proceedings of the 11th National Conference on Artificial Intelligence, Washington, DC, p. 139-144, 1993.
- [Song et al, 97] Song, W., Zabinsky, Z., Storch, R., « An algorithm for scheduling a chemical processing tank line », Production Planning and Control 35, p.

-
- 277-284, 1997.
- [Subai et al, 03] Subai, C., Niel, E., Baptiste, P., «Vers un pilotage propre des lignes de traitement de surfaces», MOSIM, France, 2003.
- [Subai et al, 06] Subai, C., Baptiste, P., Niel, E., «A Scheduling issues for environmentally responsible manufacturing: The case of hoist scheduling in an electroplating line», *Int. J. Production Economics* 99, p. 74-87, 2006.
- [Tagina et al, 09] Tagina, M., Saddam, R. et Bouabdallah, B.S., «A Multi-agent Architecture for Sensors and Actuators' Fault Detection and Isolation in Case of Uncertain Parameter Systems», *Tools and Applications with Artificial Intelligence, Studies in Computational Intelligence, Vol 166/2009*, p. 135-147, 2009.
- [Talbi, 99]. Talbi, E., «Métaheuristiques pour l'optimisation combinatoire multiobjectif», Tutorial, Journées évolutives Trimestrielles, Paris, (1999).
- [Tamaki, 92] Tamaki, H., «Maintenance of diversity in a genetic algorithm and application to the job shop scheduling», In *Proceedings IMACS/SICE Int Symp, MRP2*, p. 869-869, 1992.
- [Tangour, 07] Tangour, F., «Ordonnancement Dynamique dans les Industries Agroalimentaires», Thèse de Doctorat, École Centrale de Lille et École Nationale d'Ingénieurs de Tunis, 2007.
- [Tsang,99] Tsang, E., «A Glimpse of Constraint Satisfaction», *Artificial Intelligence Review*, 13, p.215-227,1999.
- [Varnier et al, 00] Varnier, C., Jeunehomme, N., «A cyclic approach for the multi-product hoist scheduling problem», In *7th International Workshop on Project Management and Scheduling*, Ed, Germany, 2000.
- [Varnier, 96] Varnier, C., «Extension du Hoist Scheduling. Problem cyclique-résolution basée sur les contraintes disjonctives en programmation logique avec contraintes», Thèse de doctorat, Université de Franche-Comté, 1996.

-
- [Vincke, 99a] Vincke, P., «Robust and neutral methods for aggregating preferences into any outranking relation», *European Journal of Operational Research*, No. 122, p. 405-412, 1999.
- [Vincke, 99b] Vincke, P., «Robust solutions and methods in decision-aid», *Journal of Multi-Criteria Decision Analysis*, No. 8, p. 181-187, 1999.
- [Watanabe et al, 05] Watanabe, M., Ida, K., Gen, M., «A genetic algorithm with modified crossover operator and search area adaption for the job shop scheduling problem», *Computers and Industrial Engineering*, Vol. 48, p.743-752, 2005.
- [Yamada, 03] Yamada. T., «Studies on Metaheuristics for Job-shop and Flow-shop Scheduling Problems», PhD. Thesis, Kyoto University, 2003.
- [Yih et al, 92] Yih, N., Yin, Y., «Crane scheduling in a flexible electroplating line: a tolerance based approach», *Journal of Electronic Manufacturing* 2, p. 137-144, 1992.
- [Yih et al, 95] Yih, Y., Ge, Y., «Crane scheduling with time window in circuit board production lines», *International Journal of Production Research* 33, 5, p.1187-1189, 1995.
- [Yih, 88] Yih, Y., «Trace driven knowledge acquisition for rule based real time scheduling systems», *Journal of Intelligent Manufacturing* 1, 4, p217-229, 1988.
- [Yih, 94] Yih, Y., «An algorithm for hoist scheduling problem», *International Journal of Production Research*, 32(3), p. 501-516, 1994.
- [Zhou et al, 03] Zhou, Z., Li, L., «Single hoist cyclic scheduling with multiple tanks a material handling solution», *Computers & Operations Research* 30, p. 811-819, 2003.
- [Zhou et al, 99] Zhou, X., Nishizeki, T., «Edge-Coloring and f-Coloring for Various Classes of Graphes, *Journal of Graph Algorithms and Applications* 3», 1999.
- [Zitzler et al, 99] Zitzler, E., et Thiele, L., «Multiobjective evolutionary algorithms : A comparative case study and the strength pareto approach», *IEEE*

Transactions on Evolutionary Computation,
3(4), p. 257-271, 1999.

Ordonnancement d'ateliers de traitements de surfaces pour une production mono-robot/multi-produits. Résolution et étude de la robustesse

Résumé

Nos travaux concernent la résolution d'un atelier réel de type job shop mono-robot/multi-produits dans les lignes de galvanoplastie. Assurer la production en quantité souhaitée et en qualité irréprochable, dans les délais imposés et en respectant les différentes contraintes temporelles et celles de priorité, constitue la complexité de ce type de problème.

L'objectif principal est de construire un ordonnancement cyclique proche de l'optimum d'un point de vue production devant inclure également des critères de robustesse.

Trois méthodes de résolution, à savoir l'algorithme de satisfaction de contraintes proposé qui conduit à un ensemble de solutions, celui-ci hybridé pour améliorer les résultats obtenus et l'algorithme génétique qui assure l'optimisation, ont été appliquées, pour construire divers ordonnancements de performances diverses. Une étude de la robustesse des ateliers de traitements de surfaces est enfin présentée.

Mots-clés : ordonnancement, optimisation, métaheuristiques, algorithme de satisfaction de contraintes, algorithmes génétiques, hybridation, ateliers type job shop mono-robot/multi-produits.

Solving a Job Shop Scheduling Problem on the line of treatment surface and Automation of production system

Abstract

The purpose of our works is the implementation of methodologies for the resolution of single-hoist multi-products job shop scheduling problems. Different approaches are proposed to solve this problem. Firstly a constraints satisfaction algorithm CSA is successfully applied to find the optimal move sequence of the hoist that minimize the cycle period. Then, the CSA is hybridized with two classical heuristics to impose a criterion of selection of tasks according to their processing times. This hybridization generates better quality of solution.

Secondly a specific genetic algorithm, using specific tasks coding and adapted crossover approach, is developed the case of single hoist multiple products (SHMP). A new tasks coding and crossover are proposed. To show the efficiency of this algorithm, the case of minimization of the cycle time of surface treatment line is studied. Compared to other algorithms, high improvements are achieved since the effective travel time of hoist is introduced.

In last part of this thesis, an approach for the characterization of the robustness of job shop scheduling problem in a surface treatment line is proposed

Key-words: scheduling, optimization, job shop, single-hoist multi-products Evolutionary algorithms, metaheuristics, constraints satisfaction algorithm, genetic algorithm, hybridation, robustness
