

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la
Recherche Scientifique



Ecole Nationale Polytechnique
Département de Génie Industriel

Mémoire du Projet de Fin d'Etudes d'Ingénieur

Thème

*Comparaison de l'impact d'un ordonnancement Job-shop et d'un ordonnancement
Open-shop sur les performances d'une cellule flexible d'assemblage.*

Présenté par :

M. Mohand Lounes BENTAHA

M. Hakim IDJIS

Dirigé par :

M. BAKALEM

Promotion : juin 2010

Remerciements

Nous tenons à remercier notre promoteur Mr M. BAKALEM , pour sa disponibilité tout au long de notre travail, pour ses précieux conseils et orientations ainsi que la mise en notre disposition des documents et moyens.

Nous exprimons aussi notre profonde reconnaissance à M^{lle} N. Ghazi de l'université Mouloud MAMMERI de Tizi-Ouzou et Mr. R. LATRECHE de l'ENP, notre école, pour avoir mis à notre disposition toute la documentation nécessaire.

Nous remercions tout particulièrement les membres de jury qui ont accepté de juger notre travail ainsi que tous les enseignants qui ont contribué à notre formation.

Enfin, Nous remercions aussi tous nos amis et collègues qui nous ont soutenu et tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Hakim

A

Mes parents

Mes sœurs

Mes amis

Et à tous ceux qui me sont chères

Moh Lounes

A

Mes parents

Mes sœurs et mes frères

Mes amis

Et à tous ceux qui me sont chères

ملخص

في هذه الأطروحة نتناول مشاكل الجدولة في محيط ورشة صناعية مرنة. الجدولة من النوع المفتوح و من النوع الخطي. الهدف هو تقليص مدة تواجد القطع في الورشة و ذلك ببرمجة مبنية على عدة قواعد و متغيرات مركبة بخط لا خطية (الجدولة من النوع المفتوح) و خطط خطية (الجدولة من النوع الثاني).

المشكلتين المدروستين هما $O_n / \tau_j / C_{\max}$ و $J_n / \tau_j / C_{\max}$. مقارنة مقارنة تجريبية محضة تعتمد على تقنيات التصنع و ذلك من خلال توليد برنامج تجريبي يتضمن عدة قواعد تسيير مقترحة لتحسين مردودية الورشة.

المفاتيح

الورشات المرنة، مشاكل الجدولة، الورشات المفتوحة، مقارنة تجريبية، تقنيات التصنع.

Résumé

Dans ce présent travail, nous traitons les problèmes d'ordonnancement du type Open-Shop et Job-Shop dans un environnement d'atelier flexible. Les problèmes étudiés sont $O_n / \tau_j / C_{\max}$ et $J_n / \tau_j / C_{\max}$. Notre objectif est la réduction du temps total de séjours des pièces dans un système (cyclique conçu autour d'un convoyeur en boucle), ceci est fait en considérant plusieurs paramètres de variation ainsi que des règles de priorité au niveau des postes, associé à des plans non linéaires non prédéterminés puis à des plan linéaires prédéterminés. Les temps de transfert entre les stations sont considérés. Du fait de la complexité NP- difficile au sens fort des problèmes étudiés, nous avons opté pour une approche purement expérimentale moyennant la simulation.

Les mots clés

Systeme Flexible de Production, Atelier Flexible, FMS, Systeme Cyclique d'assemblage, Ordonnancement, Open-Shop, Job-Shop, Cmax, Simulation, Approche Experimentale.

Abstract

In this paper, we study the scheduling problems in flexible manufacturing system (FMS) precisely in circular cyclic assembly system. Our objective is to minimize the Total Completion Time of jobs in this system by consideration of several parameters and rules of priority on the level of the stations, associated with not predetermined nonlinear plans then with predetermined linear plans. The scheduling problems considered are $O_n / \tau_j / C_{\max}$ and $J_n / \tau_j / C_{\max}$. Seeing the strong Np-hard complexity of the problems considered, an experimental approach is developed and validated by simulation results.

Keywords

Scheduling, FMS, Cyclic Assembly System, Cmax, Open-Shop, Job-Shop, Experimental Approach, Simulation.

Table des matières

Introduction générale	1
CHAPITRE I : GENERALITES ET ETAT DE L'ART	4
I. Introduction.....	5
II. Généralités sur les systèmes flexibles de production SFP.....	5
II.1. Le Concept CIM.....	5
II.1.1. La production	5
II.1.2. Le système de production.....	6
II.1.3. L'intégration.....	8
II.2. Les SFP	10
II.2.1. Définition d'un SFP	10
II.2.2. Les éléments constitutifs d'un SFP	10
II.2.3. La classification des SFP.....	13
III. Généralités sur le pilotage des SFP et sur l'ordonnancement	14
III.1. Pilotage des systèmes de production	14
III.2. Décomposition de la fonction pilotage	15
III.3. La conduite des systèmes de production	15
III.4. Les problèmes d'ordonnancement	17
III.4.1. Définition	17
III.4.2. La problématique d'ordonnancement	18
III.4.3. Les critères de performance et objectifs d'un problème d'ordonnancement	19
III.4.4. Les notions fondamentales d'un problème d'ordonnancement	20
III.5. Le processus global d'ordonnancement	22
III.6. Les organisations d'ateliers : (OS et JS)	23
III.6.1. Les ateliers os et js	24
III.6.2. Définition	24
III.6.3. Quelques méthodes de résolution existantes	25
IV. Conclusion	28
CHAPITRE II : MODELISATION DU SYSTEME D'ASSEMBLAGE	29
I. Introduction.....	30
II. La modélisation du système étudié.....	30
III. La description du système cyclique d'assemblage (un exemple a 4 stations)	32
IV. L'analyse du système d'assemblage	33
V. La simulation des systemes de production	37
V.1 La modélisation pour la simulation des systèmes de production	37
V.2 Définitions de la simulation	38

V.3 Les domaines d'application de la simulation	40
V.4 Les étapes des études de simulation	41
V.5 Les avantages et inconvénients de la simulation	42
VI. présentation du logiciel arena 12	43
VI.2. L'application vba de microsoft sous arena.....	46
VI.3. Illustration (model 3- 1)	47
VII. Conclusion	55
CHAPITRE III : MODELE DE SIMULATION	56
I. Introduction.....	57
II. Rappel du probleme.....	57
III. Mise en œuvre du processus de simulation.....	60
III.1. Modélisation de la cellule d'assemblage	60
III.1.1. Modélisation par l'approche processus.....	60
III.2. Construction du modèle Arena et transcription informatique du modèle.....	63
III.2.2. Interface d'expérience :.....	69
III.2.3. Interface des résultats :.....	71
III.2.4. Modélisation dynamique de l'atelier d'assemblage.....	74
III.3. Vérification et validation du modèle.....	75
IV. Conclusion.	76
CHAPITRE IV : SIMULATION ET TESTS DES DIFFERENTES STRATEGIES D'ORDONNANCEMENT PROPOSEES	78
I. Introduction :.....	79
II. La planification stratégique et tactique de la simulation :.....	79
III. Les indicateurs de performance à analyser :	80
IV. L'analyse et interprétation des résultats :.....	81
IV.1. Impact des ordonnancements Open-Shop et Job-Shop sur la cellule flexible	81
IV.2. Impact des règles de priorité au niveau des stations dans le cas Open	93
IV.3. L'approche collaborative	101
V. Conclusion.....	103
Conclusion générale.....	104
Liste des annexes.....	106
Annexe I	106
Annexe II	109
Bibliographie.....	111

1. Les ouvrages.....	111
2. Les revues.....	112
3. Les mémoires.....	113

Liste des figures

Figure I.1 : Représentation d'un système de production.....	7
Figure I.2 : Composantes du CIM.....	8
Figure I.3 : La technologie CIM.....	9
Figure I.4 : Composants de la cellule d'assemblage.....	10
Figure I.5 : schéma d'une station.....	11
Figure I.6 : Convoyeur avec accumulation.....	11
Figure I.7 : Vue complète d'une cellule flexible de production.....	12
Figure I.8 : les fonctions de pilotage.....	15
Figure I.9 : Le modèle conceptuel du système de décision.....	16
Figure I.10 : Les relations entre les différents critères.....	20
Figure I.11 : Processus global d'ordonnancement.....	22
Figure I.12 : Schéma montrant les liens entre diverses organisations.....	24
Figure I.13 : Algorithme 1 (Structure générale d'un algorithme de liste (cas sans retard)...	25
Figure I.14 : Algorithme 2 (Algorithme de liste LPT).....	26
Figure II.1 : Architecture du système d'ordonnancement basé sur la coopération	31
Figure II.2 : Exemple de cellule flexible d'assemblage étudié.....	33
Figure II.3 : Fonctionnement du système selon l'ordonnancement choisi.....	34
Figure II.4 : Cas 1 : Arrivée d'une pièce ne disposant pas d'engagement	35
Figure II.5 : Cas 2 : Arrivée d'une pièce disposant d'un engagement	36
Figure II.6 : Domaine d'application de la simulation.....	40
Figure II.7 : Processus simplifié de simulation.....	41
Figure II.8 : Processus de simulation selon Pritsker.....	41
Figure II.9 : Interface graphique d'Arena 12 pour model 3-1.....	44
Figure II.10 : La barre d'outils.....	45
Figure II.11 : Création du fichier.....	47
Figure II.12 : Module <i>Create</i>	47
Figure II.13 : Module <i>Process</i>	48
Figure II.14 : Module <i>Dispose</i>	48
Figure II.15 : le modèle.....	49
Figure II.16.a : Module <i>Entity</i>	49
Figure II.16.b : Module <i>Entity</i>	49
Figure II.17 : Module <i>Queue</i>	50
Figure II.18 : Module <i>Resource</i>	50
Figure II.19.a : Animation des variables.....	51
Figure II.19.b : Animation des variables.....	51
Figure II.20 : La fenêtre <i>Setup</i>	52
Figure II.21 : Le modèle après exécution.....	53
Figure II.22 : Le fichier Rapport.....	53
Figure II.23 : Exemple de Script VBA sous.....	54
Figure II.24 : Exemple d'exécution du script.....	54
Figure II.25 : Les moments d'intervention de VBA.....	55

Figure III.1 : Philosophie générale du protocole de coopération.....	58
Figure III.2 : Vue complète du modèle.....	63
Figure III.3 : Modélisation de l'entrée du système.....	64
Figure III.4 : Modélisation d'une station par un <i>submodel</i> (station 1).....	65
Figure III.5 : contenu d'un <i>submodel</i> (station 4).....	66
Figure III.6 : Modélisation de la station.....	67
Figure III.7 : Modélisation de la sortie d'une station.....	68
Figure III.8 : Modélisation de la sortie du système.....	68
Figure III.9 : Interface d'expérience.....	70
Figure III.10 : Changement du Tag de la variable <i>choose</i>	70
Figure III.11 : Interface de résultats.....	72
Figure III.12 : La représentation graphique de la cellule.....	74
Figure III.13 : a. Station, b. file d'attente d'entrée des stations, c. File d'attente de sortie..	74
Figure III.14 : modélisation des Palette par des transporteurs.....	75
Figure III.15 : le système en plein évolution (6 stations).....	75
Figure IV.1 : Impact de la taille de commande sur le <i>Cmax</i>	83
Figure IV.2 : } Impact de la capacité des files d'attente sur le <i>Cmax</i>	84
Figure IV.6 : }	
Figure IV.3 : } Impact de la longueur du convoyeur sur le <i>Cmax</i>	85
Figure IV.7 : }	
Figure IV.4 : } Impact du nombre de palettes sur le <i>Cmax</i>	86
Figure IV.8 : }	
Figure IV.5 : }	
Figure IV.9 : }	
Figure IV.10 : }	
Figure IV.11 : } Impact des différentes organisations sur le <i>Cmax</i>	87 - 91
Figure IV.12 : }	
Figure IV.13 : }	
Figure IV.14 : }	
Figure IV.15 : Impact de toutes les expériences sur le <i>Cmax</i>	91
Figure IV.16 : Taux d'amélioration du <i>Cmax</i> selon la taille de la commande.....	94
Figure IV.17 : Taux d'amélioration du <i>Cmax</i> toute taille confondues.....	94
Figure IV.18 : Taux d'amélioration du <i>Cmax</i> selon la capacité des files d'attente.....	95
Figure IV.19 : Taux d'amélioration du <i>Cmax</i> selon la distance entre stations.....	95
Figure IV.20 : Taux d'amélioration du <i>Cmax</i> selon le nombre de palettes.....	96
Figure IV.21 : }	
Figure IV.22 : }	
Figure IV.23 : }	
Figure IV.24 : } Impact des différentes règles sur le <i>Cmax</i>	96-99
Figure IV.25 : }	
Figure IV.26 : }	
Figure IV.27 : }	
Figure IV.28 : Moyennes des taux d'amélioration.....	99
Figure IV.29 : Ecart type des taux d'amélioration.....	99

Figure IV.30 : Impact des règles de priorité (toutes les expériences).....	100
Figure VI.31 : Impact de la capacité des files d'attente.....	101
Figure VI.32 : Impact de la distance entre stations.....	101
Figure VI.33 : Impact du nombre de palettes.....	102
Figure VI.34 : Impact de l'approche collaborative sur le <i>C_{max}</i>	102

Liste des tableaux

Tableau IV.1 : Planification des expériences.....	80
Tableau AI : La notation en $\alpha/\beta/\gamma$ des problèmes d'ordonnancement.....	108

Liste des abréviations

AC: Area Controller.
AGV: Automated Guided Vehicle.
AMT: Advanced Manufacturing Technology.
ASRS: Automated Storage Retrieval Systems.
CAD: Computer-Aided Design.
CADD: Computer-Aided Design and Drafting.
CAE: Computer-Aided Engineering.
CAM: Computer-Aided Manufacturing.
CAP: Computer- Aided Planning.
CAPP: Computer- Aided Process Planning.
CAQ: Computer- Aided Control.
CC: Cell Controller.
CIM: computer integrated manufacturing.
C_{max} : Date de fin de la dernière pièce traitée.
CNC: Computer Numerical Control.
DAS: Data Acquisition systems.
DNC: Direct Numerical Control.
Doi : Durée opératoire d'une pièce sur la station *i*.
EDD: Earliest deadline.
EDI: Echanges de Données Informatiques.
EG : Engagement.
EOJ: End Of Job.
EOM: End Of Machine.
FAEi : File d'attente d'entrée.
FAS: Flexible assembly system.
FIFO: First In First Out.
FMC: flexible manufacturing cell.
FMS: flexible manufacturing system.
FS: Flow-Shop.
SFP: Système Flexible de Production.
GPAO: Gestion de la Production Assisté par Ordinateur.

JAT: Just In Time.

JS: Job-Shop.

LAPT: Longest Alternate Processing Time first.

LPT: Longest Processing Time.

LWKR: Least Work Remaining first.

MDA: machine data acquisition.

MOCN : machine outil à Commande Numérique.

MRP: Materials Resource planning.

MWKR: Most Work Remaining first.

NC: Numerical Control.

OS: Open-Shop.

PD : Les méthodes basées sur la Programmation Dynamique.

PDA: production data acquisition.

PL : Les méthodes basées sur la Programmation Linéaire.

PSE : Les Procédures par Séparation et Évaluation.

SID: système d'information et de décision.

SMA : Système Multi Agents.

SPT: Shortest Processing Time.

TQM: Total Quality Management.

Introduction générale

L'évolution du monde industriel est continue. Elle est fonction des besoins croissants et variés des clients et doit répondre à l'augmentation de la compétitivité inhérente à la mise en œuvre de l'économie. Désormais, le modèle des Systèmes de Production (SP) centralisés n'a plus la place stratégique qu'il occupait dans les anciens contextes industriels car il doit pouvoir supporter un processus de fabrication flexible, réactif et économique. La flexibilité dont il est question ici, se traduit par une facilité de réadaptation et de réaffectation des fabrications variant dans le temps, et ceci de manière compétitive. Ainsi, les systèmes de production correspondants doivent être associés à des systèmes de pilotage adaptables et autonomes. Ces derniers doivent rester ouverts, c'est-à-dire évolutifs et capables de supporter de manière dynamique une reconfiguration ou un réajustement des paramètres opérationnels tant au niveau du produit, du procédé que du marché.

Le développement des systèmes de production automatisés, associés à des outils de supervision (les systèmes flexibles de production) a conduit à l'émergence de la fonction pilotage de l'unité de production. Piloter l'unité, c'est d'une part communiquer au système de commande un ensemble de paramètres lui permettant de fonctionner et d'autre part réagir à la réponse du système ainsi qu'aux aléas.

Un processus de production peut être vu comme une alternance d'opérations de production, effectuées sur des postes de travail et d'opérations de transport, souvent importantes. L'étude des systèmes de transfert automatisés s'est beaucoup développée avec l'apparition des Systèmes Flexibles de Production (SFP).

Dans ce présent travail, on se propose d'étudier les problèmes d'ordonnancement du type $O_n / \tau_j / C_{\max}$ et $J_n / \tau_j / C_{\max}$ dans un environnement d'un système de production flexible, dont le moyen de transport est un convoyeur en boucle. On s'intéressera aux impacts de ces derniers sur les performances de la cellule flexible étudiée, tout en les comparant avec celles enregistrées pour le Flow Shop (FS) de mêmes dimensions. Cette étude est basée sur la minimisation des temps d'assemblage d'une commande de pièces (C_{\max}) dans le système.

Une pièce est obtenue par transformations successives sur un sous-ensemble des stations du système. L'ordre dans lequel les stations sont visitées par une pièce est appelé routage. Dans la pratique, les routages sont choisis de façon à équilibrer les charges des stations, ce choix est fait avant de résoudre le problème d'ordonnancement que nous posons ici. Si l'on ajoute au routage les temps de passage des pièces sur les machines, on dit que l'on connaît la gamme de fabrication des pièces. Lorsque le routage est le même pour toutes les pièces à fabriquer, on dit que l'on a affaire à une production linéaire, ou Flow-Shop. Dans le cas contraire, on parle de production non linéaire, ou Job-Shop, et si en plus il n'est pas connu on parle alors d'Open-Shop (les contraintes de précedence entre les opérations d'une gamme de

production sont éliminées). C'est à ces deux derniers types de production qu'on s'intéresse dans le cadre de cette étude.

Une étude sur l'apport d'un arrangement circulaire aux performances des lignes d'assemblage linéaires a été effectuée par Finke et Dupont. Ils ont conclu que les performances d'un système d'assemblage cyclique sont conditionnées par le robot goulot et que son pilotage par des plans non linéaires (Job-shop) n'apportait aucune amélioration aux performances du système, mais que l'intérêt est dans la manipulation des plans d'assemblage ouverts d'où l'acquisition d'une flexibilité dans l'ordre de passage des tâches (l'Open-shop), où chaque ordre des opérations d'assemblage est possible. Ils ont montré que l'apport d'une combinaison plans non linéaire-pilotage FIFO à la performance du système est peu signifiant (inférieur à 5%).

A partir de ces constats, on se propose d'étudier et d'analyser le comportement d'un tel système géré d'abord par des gammes prédéterminées c'est-à-dire une organisation Job-Shop, en suite par des gammes non prédéterminées c'est-à-dire une organisation Open-Shop. On applique à chacune des deux organisations dans un premier temps un ordonnancement par la règle de priorité FIFO (First In First Out), puis on s'intéressera au cas Open-Shop combiné a des règles de priorité au niveau des stations de travail.

L'objectif de l'étude est de comparer les performances de la cellule d'assemblage d'abord sous une organisation job-Shop en suite sous une organisation Open-Shop tout en vérifiant les résultats avancés par l'étude effectuée par Finke et Dupont (Finke et Dupont, 1993) ainsi que Ghazi (Ghazi, 2007). Le système étudié est une cellule flexible d'assemblage dont les stations sont conçues autour d'un convoyeur en boucle avec accumulation. Les paramètres de variation sont:

- Nombre de palettes ;
- Nombre de stations ;
- Capacité des files d'attente d'entrée ;
- Longueur du convoyeur ;
- Les règles de pilotage : LPT, SPT, LRWK et MRWK.

Vu la complexité des problèmes considérés, l'approche de résolution est une approche purement empirique. C'est-à-dire, l'étude sera établie sur un programme expérimental pré-généré. Pour ce faire, on optera pour un outil d'aide à la décision à savoir la simulation par RockWell ARENA 12.

Ce mémoire comporte quatre chapitres. Dans le premier chapitre ainsi que le deuxième, on abordera toutes les notions élémentaires nécessaires à notre étude. Le troisième chapitre ainsi que le quatrième sont consacrés à la partie pratique de cette étude.

Le premier chapitre comporte deux grandes sections :

Introduction générale

Dans la première section, on présentera les systèmes flexibles de production SFP, dans la deuxième, on développera des généralités sur le pilotage de ces derniers ainsi que les notions essentielles des problèmes d'ordonnancement d'atelier.

Comme le premier chapitre, le deuxième comporte deux grandes sections :

La première est une description détaillée du système, ainsi que la présentation de notre approche tout en justifiant l'utilisation de la simulation comme outil de résolution. La seconde introduit les notions de base de la simulation avec les neuf étapes de construction d'un projet de simulation, définies par (Pritsker, 1986).

Le troisième chapitre est une mise en œuvre des six premières étapes (présentation du modèle établi).

Le quatrième et dernier chapitre est réservé pour les trois dernières étapes du processus de simulation. Nous allons simuler et tester les différentes stratégies d'ordonnancement proposées.

Ce mémoire est accompagné de deux annexes : la première est un glossaire avec un tableau détaillé de notations des problèmes d'ordonnancement, la seconde est un résumé de l'approche processus (approche de simulation).

Nous concluons ce mémoire par une conclusion générale, dans laquelle nous développons des perspectives issues des résultats obtenus de notre travail.

Chapitre I

GENERALITES ET ETAT DE L'ART

*“La science restera toujours la satisfaction du plus
haut désir de notre nature ; la curiosité, elle
fournira à l'homme le seul moyen
qu'il ait d'améliorer son sort”*

[Ernest Renan]

Chapitre I : Généralités et Etat de l'art

- I. Introduction
- II. Généralités sur les systèmes flexibles de production **SFP**
 - II.1 Le concept CIM (Computer Integrated Manufacturing)
 - II.1.1. La production
 - II.1.2. Le système de production.
 - II.1.3. L'intégration
 - II.2. Les Systèmes Flexibles de Production
 - II.2.1. Les définitions d'un SFP
 - II.2.2. Les éléments constitutifs d'un SFP
 - II.2.2. 1. Les stations de travail
 - II.2.2.2. Les équipements de transport
 - II.2.2.3. Les systèmes de stockage
 - II.2.2.4. Les systèmes de contrôle et de communication
 - II.2.3. La classification des SFP
 - II.2.3.1. Les ateliers flexibles spécialisés
 - II.2.3.2. Les ateliers flexibles complexes
 - II.2.3.3. Les cellules ou îlots flexibles
- III. Généralités sur le pilotage des SFP et sur l'Ordonnancement
 - III.1. Le pilotage des systèmes de production
 - III.2. La décomposition de la fonction pilotage
 - III.3. La conduite des systèmes de production
 - III.4. Les problèmes d'ordonnancement
 - III.4.1 Définition
 - III.4.2 La problématique de l'ordonnancement
 - III.4.3 Les critères de performance d'ordonnancement
 - III.4.4 Les notions fondamentales d'un problème d'ordonnancement
 - III.5. Le processus global d'ordonnancement
 - III.6. Les organisations d'ateliers
 - III.6.1. Les ateliers JS et OS
 - III.6.2. Définition
 - III.6.3. Quelques méthodes de résolution existantes
 - III.6.4. La complexités des problèmes d'ordonnancement
- IV. Conclusion

I. Introduction

Les modifications, améliorations ou transformations de produits, sont de plus en plus fréquentes et la notion de flexibilité est fortement liée à cette évolution. La concurrence exige que l'entreprise se dote d'un système de production efficace, qui réagisse rapidement aux contraintes du monde environnant et en particulier aux exigences et évolutions du marché.

Deux types d'évolutions non antagonistes, bien au contraire, semblent émerger pour tenter de répondre au problème de la survie de l'entreprise :

- intégration cohérente de la gestion de production avec une importance particulière pour la transmission des informations et de la communication.
- automatisation de la fabrication pour l'amélioration de l'exécution des tâches et de l'ensemble des flux physiques.

La première section de ce chapitre est consacrée à la partie physique du système de production et plus précisément aux systèmes flexibles de production (SFP). La deuxième section est consacrée à l'introduction du concept de la productique (CIM) et les systèmes flexibles de production (SFP). La troisième section quand à elle est consacrée à la partie pilotage des systèmes de production et plus précisément à l'ordonnancement où on focalisera notre attention sur les éléments fondamentaux d'un problème d'ordonnancement. Et en dernier, on s'intéressera aux différentes organisations d'ateliers afin d'introduire les cas considérés dans notre étude : l'Open-Shop et le Job-Shop.

II. Généralités sur les systèmes flexibles de production SFP

II.1. Le concept CIM

II.1.1. La production

a). Définition

La **production** est une transformation de ressources appartenant à un système productif et conduisant à la création de biens *ou* de services (Giard, 2003).

La production d'un bien s'effectue par une succession d'opérations consommant des ressources et transformant les caractéristiques morphologiques de «matières» (au sens large du terme) ou leur localisation (manutention/transport). La production d'un service s'effectue elle aussi par une succession d'opérations consommant ces ressources sans qu'il y ait nécessairement transformation de matières. Les ressources mobilisées dans le processus de production peuvent être de quatre types:

- des équipements (bâtiments, machines, outillage, etc.);
- des hommes (opérateurs intervenant soit directement dans le processus de transformation, soit indirectement, ce qui correspond à des activités dites de support);
- des matières (matières premières, composants, etc.);
- des informations techniques ou procédurales (gammes, nomenclatures, procédures, etc.).

Gérer une production consiste – entre autres – à organiser les flux physiques de produits au travers de moyens de production.

b). Typologie de production

Chaque entreprise est unique par son organisation et par la spécificité des produits qu'elle fabrique. Cependant, on peut réaliser la classification suivante (Courtois et al, 2002).

- Quantités fabriquées et répétitivité ;
 - Production unitaire ;
 - Production par petites séries ;
 - Production par moyennes séries ;
 - Production par grandes séries.
- Organisation des flux de production ;
 - Production en continu ;
 - Production en discontinu ;
 - Production par projet.
- Relation avec les clients.
 - Vente sur stock ;
 - Production à la commande;
 - Assemblage à la commande.

II.1.2. Le système de production

Un système de production est un ensemble de ressources réalisant une activité de production (Giard, 1988).

Une des fonctions importantes du système de production est la fabrication elle-même du produit fini, mais son bon déroulement nécessite la mise en œuvre de fonctions additionnelles telles que les achats de composants et de matières premières, la distribution du produit fini, la gestion de la qualité des composants et du produit où la maintenance des différentes ressources intervienne aussi de manière importante.

a). Décomposition du système de production

Les systèmes de production peuvent être des systèmes très complexes et difficiles à gérer. De ce fait, ils sont beaucoup étudiés, et ce depuis longtemps. Plusieurs approches ont été envisagées dans le but de mieux comprendre leur fonctionnement et de mieux les appréhender.

D'une manière classique, tout système de production est décomposé, selon une approche systémique, en plusieurs sous systèmes afin de faciliter la maîtrise de la complexité associée à la fonction production. Dans la littérature, plusieurs auteurs ont proposé différentes décompositions.

D'un point de vue systémique (Le Moigne, 1990), le système de production représente le système opérant du système « entreprise manufacturière ». Selon ce point de vue, il est classique de décomposer le système *entreprise* en trois sous-systèmes qui coopèrent :

- *Le système physique* représente le système opérant du système de production ;
- *Le système d'information* permet l'acquisition, le traitement et la gestion des données du système de production et ses environnements internes et externes. Ces données permettent de connaître la situation du système à tout instant ;
- *Le système de décision* identifie, analyse et corrige les dérives du système physique, il le pilote .Ses interventions se font en fonction des objectifs assignés qui constituent une contrainte dont il doit tenir compte.

Si cette décomposition est valable pour le système *entreprise*, et permet son analyse, elle est beaucoup moins adaptée pour les systèmes de production et sa modélisation. En effet, dans le système de production, les sous-systèmes d'information et de décision n'ont pas d'existence propre, l'un sans l'autre. Ils constituent ensemble ce que nous appelons le système de pilotage ou le système d'information et de décision (SID) (figure 1.1) ou encore le système directeur (Rodde, 1991) ou la partie conduite (Draghici, 1998). Ainsi, il est plus conforme à la réalité de considérer le système de production comme l'association d'un système physique et d'un système de pilotage (Bakalem, 1996) ou encore d'une partie opérative et d'une partie commande (figure I.1).

L'association des parties des systèmes de décision et d'information concernant uniquement la production constitue le système de gestion de production (Letouzey, 2001).

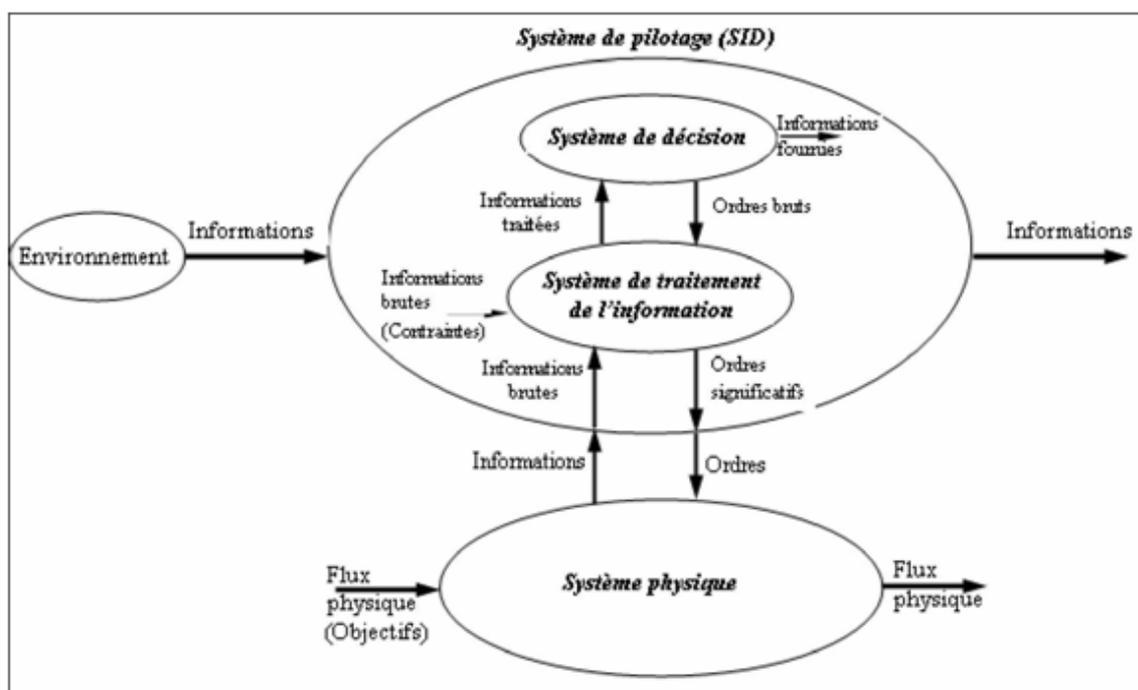


Figure I.1 : Représentation d'un système de production (Bakalem, 1996)

b). Caractéristiques des systèmes de production

Si la valeur d'un produit était autrefois caractérisée par son coût, elle intègre désormais d'autres aspects tels que : la qualité, la possibilité de personnalisation (créer la diversité), les délais de livraison,...etc. Pour répondre à ces nouvelles exigences, les systèmes de production doivent se doter de nouvelles caractéristiques telles que : la flexibilité, la réactivité, la proactivité et la robustesse.

Flexibilité : « *La flexibilité est la capacité ou l'aptitude ou la rapidité d'un système industriel à créer et à gérer la variété, de façon économique et continue, afin de s'adapter aux changements de l'environnement, externe ou interne, tout en maintenant son équilibre (sa stabilité)* » (Kieffer et Pujo, 2002).

Réactivité : « *La réactivité est la capacité du système à élaborer une solution admissible face à un aléa, dans un délai suffisant par rapport à la dynamique de l'environnement* » (Kieffer et Pujo, 2002).

Robustesse : la robustesse est définie comme suit : « *La robustesse d'un système de production est sa capacité à maintenir ses performances, face aux perturbations.* » (Beslon, 1995).

Proactivité : « *La proactivité d'un système de production se caractérise par ses capacités d'anticipation (prévoir et/ou provoquer) les changements d'état, d'apprentissage et d'enrichissement des connaissances (pour améliorer sa réactivité), d'adaptation de ses règles de fonctionnement et par sa capacité de réorganisation reposant sur une architecture décentralisée et une délégation de responsabilité.* » (Ghazi, 2007)

II.1.3. L'intégration

Jusqu'à récemment, l'intégration de l'information et le transfert de données a été toujours manuel, habituellement en utilisant le papier comme moyen de transfert. Avec l'avènement des ordinateurs et les technologies de communication, il a été possible d'intégrer des activités de production sans avoir recours au papier. L'intégration de ces activités (marketing, investissement, planification, achat, vente,) à travers le transfert automatique de données est le but du CIM, qui préconise d'automatiser et puis intégrer les activités directes et indirectes d'une usine.

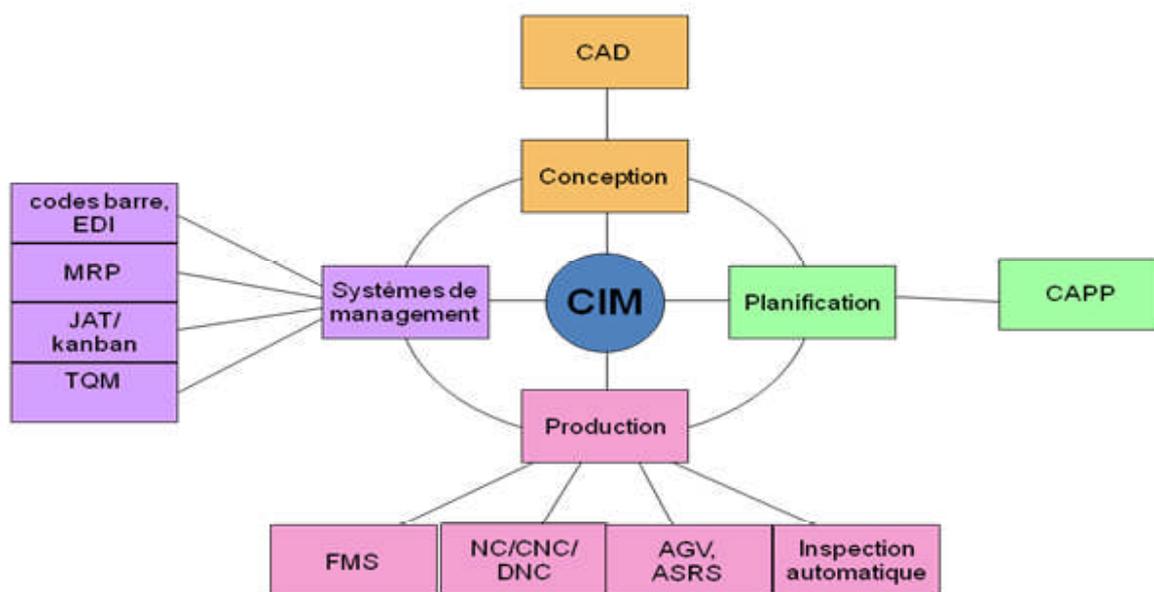


Figure I.2 : Composantes du CIM

L'automatisation des activités directes de production et quelques unes des indirectes est connue sous le nom de *Advanced Manufacturing Technology (AMT)* ou *Technologie de Production Avancée*, qui s'occupe de l'automatisation et l'intégration de toutes les activités d'une entreprise qui concernent les opérations de production comme la conception, la production, la qualité, etc. Les activités organisationnelles comme la finance, le marketing, etc., ne relèvent pas de l'AMT, cette dernière englobe seulement l'automatisation des activités d'engineering et de production comme le *Computer-Aided Manufacturing (CAM)* en français la *Gestion de la Production Assisté par Ordinateur (GPAO)* ou le *Computer-Aided Engineering (CAE)* en français l'*Engineering Assisté par Ordinateur*.

Le CAE englobe:

1. Le CAD. Computer- Aided Design ou Conception Assisté par Ordinateur (CAO).
2. Le CAQ. Computer- Aided Control ou Contrôle Assisté par Ordinateur.
3. Le CAP. Computer- Aided Planning ou Planification Assisté par Ordinateur.

Le CAM englobe:

1. Le DAS. Data Acquisition systems qui englobe MDA (machine data acquisition) PDA (production data acquisition). MDA est la collecte automatique et l'analyse des données fournies par les équipements, PDA utilise les données de MDA et celles fournies par les opérateurs (personnel data)
2. Le DNC. Direct(ou distributed) numerical control ou la commande numérique directe (plusieurs machines-outils sont commandées par un ordinateur central)
3. Le FMS. Flexible manufacturing system ou système flexible de production qui est constitué d'un groupe d'équipements sous le contrôle de l'ordinateur central du SFP.
4. Le FAS. Flexible assembly system ou système flexible d'assemblage.

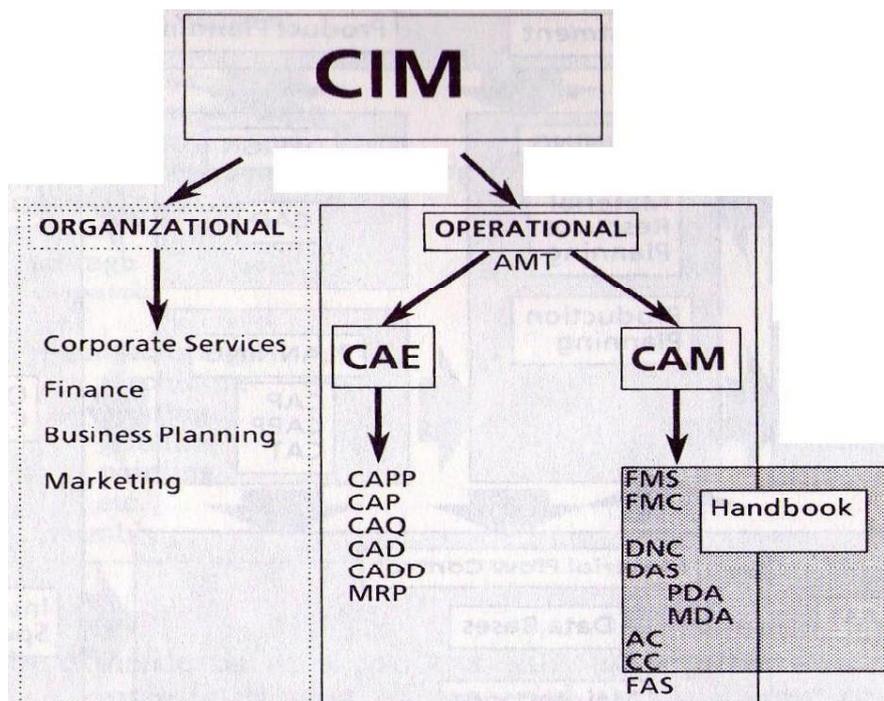


Figure I.3 : La technologie CIM (Parrish, 1993)

II.2. Les SFP

II.2.1. Définition d'un SFP

« Un SFP est un groupe de machines de production programmables intégrées avec un équipement de manutention automatique sous le contrôle d'un ordinateur centrale destiné à la production d'une variété de produit avec des cadences, tailles des lots et quantités non uniformes (Nand, 1991).

D'après (Parrish, 1993) « Un SFP est un groupe d'équipements de production logiquement organisés sous un ordinateur central et physiquement connectés par un système de transport. »

La définition englobant les points importants est celle de (Jubin, 1994) :

« Un atelier flexible permet la production automatique de pièces de types divers et en quantités variables. Les opérateurs n'interviennent pas directement dans le processus de fabrication et limitent essentiellement leurs interventions à l'entretien. L'ordonnancement est géré par un système informatique. »

II.2.2. Les éléments constitutifs d'un SFP

Les systèmes flexibles de production sont constitués d'un certain nombre d'équipements qui peuvent être divisés en quatre familles : (Ghazi, 2007)

- Les stations de travail ;
- Les systèmes de manutention et de transport ;
- Les systèmes de stockage ;
- Les systèmes de contrôle et de communication.

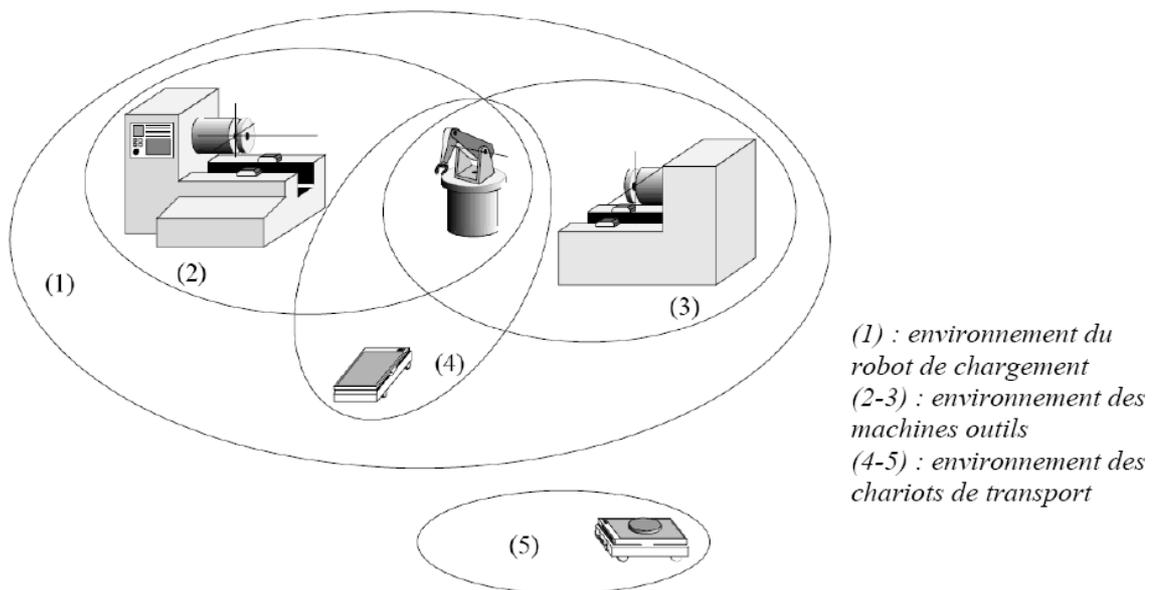


Figure I.4: Composants de la cellule d'assemblage (Beslon, 1995).

II.2.2. 1. Les stations de travail

Une station de travail est composée d'une MOCN, d'un magasin d'outils avec un système automatique de changement d'outils, d'un magasin à palettes et supports, d'un système automatique de chargement/déchargement des palettes dans les machines, d'un système automatique d'inspection et d'un système de contrôle.

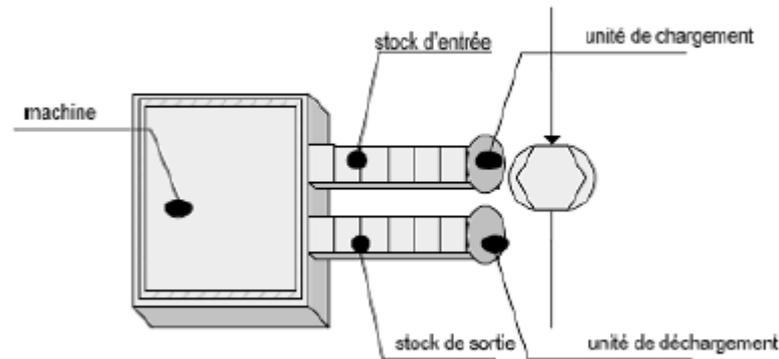


Figure I.5 : schéma d'une station (Caumon, 2006)

II.2.2.2. Les équipements de transport

Les équipements de transport servent à déplacer les produits et éventuellement les outils entre les machines suivant les chemins planifiés pendant la conception du FMS. On peut citer les convoyeurs, les véhicules industriels, les véhicules autoguidés, ...etc.



Figure I.6 : Convoyeur avec accumulation (Ghazi, 2007).

II.2.2.3. Les systèmes de stockage

Les aires de stockage peuvent aller du système complètement automatique au stockage totalement manuel. Une aire de stockage est composée d'un bâtiment, d'un système de rangement (racks, étagères...) d'un mécanisme de transport, de chargement et déchargement et d'un système de contrôle.

II.2.2.4. Les systèmes de contrôle et de communication

Le système de contrôle peut être composé d'un ordinateur central de commande, il donne des instructions et reçoit des états de situations de chaque équipement du SFP. C'est le système de contrôle qui décide quand et/ou comment les produits doivent se déplacer entre les différentes machines et être chargés ou déchargés. Un système de commande élaboré peut commander et ordonnancer en temps réel le SFP.

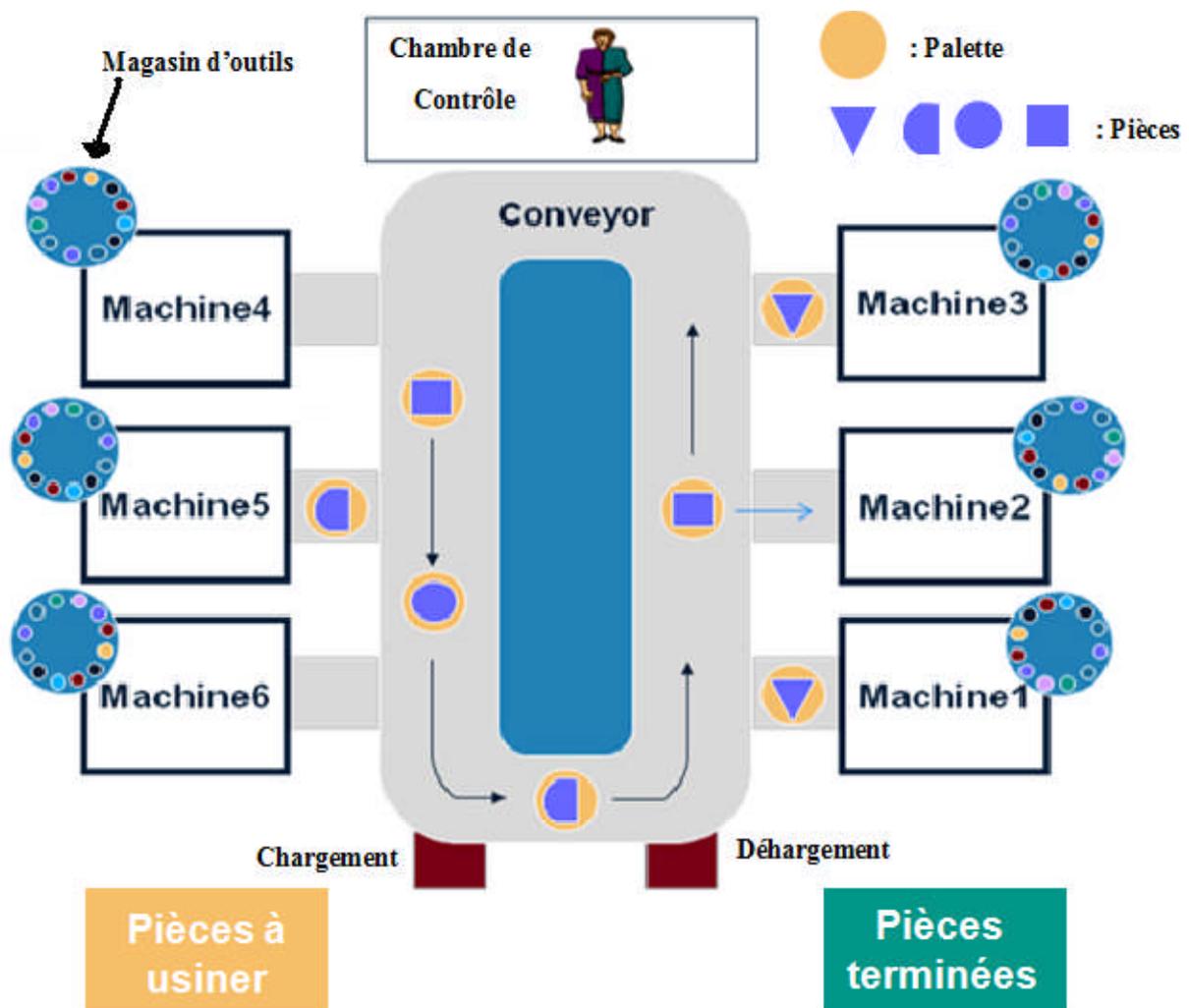


Figure I.7 : Vue complète d'une cellule flexible de production

II.2.3. La classification des SFP

Il existe de nombreuses classifications des différents types de systèmes flexibles de production. Les critères de classification considérés par les auteurs sont variés. Citons par exemple la méthode de fabrication utilisée, la taille des machines et leur complexité, les modèles des flux de pièces à traiter ainsi que les caractéristiques de fonctionnement et de contrôle du système, la topologie de l'atelier...etc. Nous retenons celle proposée par (Jubin, 2003) : les ateliers flexibles spécialisés ou lignes de transferts flexibles, les ateliers flexibles complexes et les cellules ou îlots flexibles.

Les ateliers flexibles spécialisés ou lignes de transferts flexibles : ces ateliers sont conçus pour la production de variantes dimensionnelles de pièces d'une même famille sur le plan fonctionnel et morphologique. A cette famille correspond une gamme unique (une succession d'opérations bien déterminées) ; les pièces passent **successivement** sur les différentes machines de l'atelier disposées en ligne dans l'ordre où elles interviennent suivant la logique du « juste à temps ». Le nombre de machine (MOCN) est généralement compris entre 4 et 10 unités.

Les ateliers flexibles complexes : Regroupant plusieurs (5 à 20) MOCN associées à des systèmes de transfert de pièces et de chargement/déchargement totalement ou partiellement automatisés selon les cas. Ces ateliers sont destinés à la production de pièces variées du point de vue de la forme et des dimensions. Aux divers types de pièces correspondent des gammes différentes, ces gammes ont cependant certaines opérations en commun de manière à ce qu'une machine donnée du système puisse intervenir dans la fabrication de plusieurs pièces de la même famille. La progression des pièces dans un atelier flexible de ce type n'est pas prédéterminée. Le logiciel de commande doit assurer une véritable **gestion en temps réel** en fonction des règles de pilotage préalablement établies.

Les cellules ou îlots flexibles : Par rapport aux ateliers flexibles, les cellules sont des systèmes moins complexes, regroupant un petit nombre (2 ou 4) MOCN. A ces machines sont associés, comme pour les ateliers, divers équipements assurant de manière automatique les fonctions de stockage, de transfert, de chargement/déchargement (des pièces brutes et finies) et de contrôle. Les machines constituant les cellules flexibles sont associées les unes aux autres en fonction de l'ensemble des opérations à réaliser indépendamment de l'appartenance des pièces à une famille bien déterminée.

La classification précédente a seulement pour objectif de situer plusieurs niveaux d'automatisation flexible. Dans la réalité, la situation peut s'avérer un peu plus complexe ; certains systèmes flexibles, exploités dans l'industrie, ont en effet des caractéristiques qui les apparentent à la fois aux cellules et à certaines formes d'ateliers flexibles.

III. Généralités sur le pilotage des SFP et sur l'Ordonnancement

III.1. Le Pilotage des systèmes de production :

Selon une optique productique, l'objectif principal de la fonction pilotage de la production est la bonne exécution du programme prévisionnel de production par le système physique (Nagi, 1991).

Le système de pilotage collecte, stocke, traite et transmet des informations d'une part, et pilote le système à travers les informations recueillies, en fonction de l'environnement et de son processus de décision, d'autre part. Il est composé :

- Des points de captage (les capteurs), pour la récupération de l'information à différents points du système de production ;
- Du processus de décision pour l'analyse et le traitement de l'information et la génération des décisions ;
- Des points d'actions qui constituent les points de passage d'ordres vers le système physique.

Le système de pilotage récupère en entrée des informations et fournit en sortie, après traitement, des informations et des décisions. Ces décisions peuvent être soit des ordres de fabrication (issus du processus d'ordonnancement) soit des actions correctives (réaction aux aléas) sur le système, impliquant la mise en œuvre de certaines règles de conduite. La réactivité du système de production dépend en premier lieu de la capacité de réaction de son système de pilotage. Cette capacité est liée au nombre et à la qualité des points de captage et des points d'action d'une part, et aux processus de décision, d'autre part. Le système de pilotage est ainsi un ensemble de points de pilotage (points de captage + processus de décision + points d'action) interférents (Bakalem, 1996).

Selon (Melnyk et Carter, 1987), le pilotage d'atelier est défini comme *le groupe d'activités directement responsable de la gestion de la transformation d'ordres de fabrication planifiés en pièces sorties de l'atelier*. Il gouverne le planning court terme détaillé, les activités d'exécution et de suivi nécessaires pour contrôler l'écoulement d'un ordre de fabrication, depuis le moment où l'ordre est lancé par le planning jusqu'à son exécution complète.

Le pilotage affecte le travail des moyens de production afin de respecter le plan et les délais fixés par la planification à court terme. Cette activité s'appelle l'ordonnancement.

Bel (Bel, 1991) complète la définition de Melnyk, en indiquant *que le pilotage d'atelier couvre les activités de conduite du système et la mesure de son état en vue de déterminer les actions correctives les plus appropriées*.

III.2. La décomposition de la fonction pilotage :

Le pilotage est défini par (Trentesaux et al, 1996) comme la structure de décision et d'information associée à la gestion temps réel. La fonction pilotage "temps réel" comprend deux sous-fonctions (figure I.8) :

- la fonction *conduite*,
- la fonction *commande*.

La fonction *conduite* regroupe les activités décisionnelles qui sont traduites en ordres et transmises au niveau commande. Ce dernier niveau gère la réalisation du processus physique de fabrication et retourne au niveau conduite un ensemble de données de suivi. Cette approche reprend la notion de boucle de rétroaction (Ghazi, 2007).

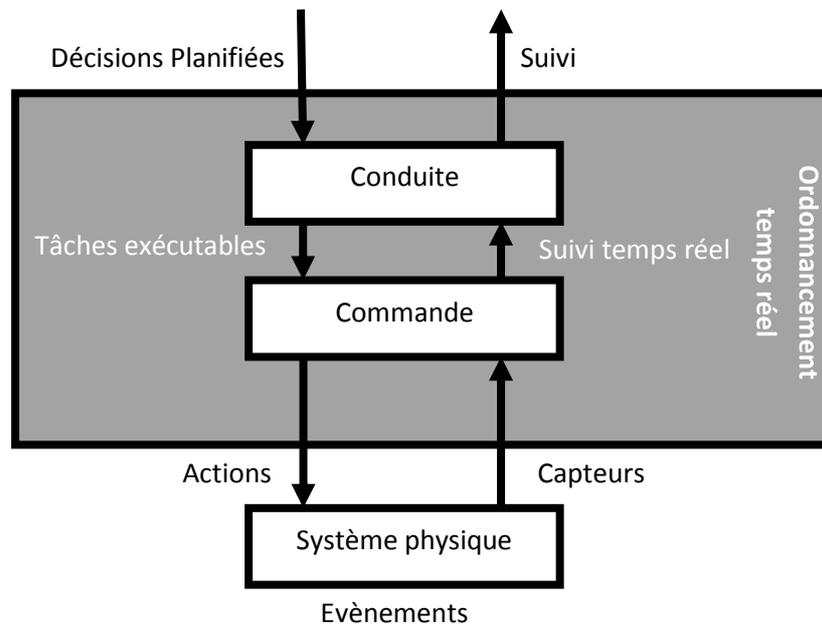


Figure I.8: les fonctions de pilotage

III.3. La conduite des systèmes de production :

La fonction principale de la partie conduite est d'élaborer les ordres nécessaires à la partie opérative (physique). Le rôle de la conduite des systèmes de production est de diriger, guider et piloter, de manière à assurer la pertinence et la cohérence du système dans un environnement donné. La conduite peut être considérée comme l'art d'adapter en permanence les objectifs de l'entreprise à l'évolution de l'environnement, à travers l'analyse des contraintes et des opportunités (Ghazi, 2007). La conduite réactive d'un système de production se définit comme la capacité à répondre à une demande fluctuante en nature et en quantité de produits variés et personnalisés, dans des délais respectés et de plus en plus courts. C'est une démarche visant à répondre à la demande du client et correspond à l'un des facteurs

mesurant la compétitivité industrielle de l'entreprise. Un système est réactif si les conditions de flexibilité sont assurées au sein de l'entreprise.

Les principales missions du système de conduite peuvent être résumées de la façon suivante : auto-organisation, adaptation, optimisation et régulation.

La conduite d'un système de production consiste à utiliser un système de décision, pour faire exécuter par le système physique l'ensemble des opérations de fabrication qui lui sont affectées :

- en respectant au mieux les objectifs de production fixés, tout en satisfaisant les contraintes spatiales, temporelles et de coûts ;
- en s'assurant que chaque ordre transmis est cohérent vis-à-vis du caractère perturbé de l'environnement dans lequel évolue le système ;
- en utilisant un système d'information cohérent réalisant une interface robuste entre le système physique et le système de décision.

Le système de décision définit des cadres de décisions dans lesquels il précise les opérations à accomplir, les grandeurs sur lesquelles il est possible d'agir, les méthodes à suivre ainsi que les limites. Il s'articule autour d'une organisation en boucle (Figure I.9) constituée de quatre activités suivantes :

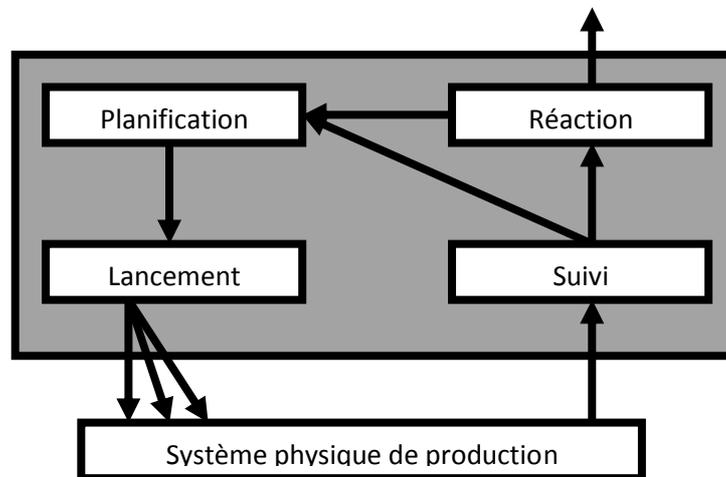


Figure I.9: Le modèle conceptuel du système de décision (Draghici et al, 1998).

► *la planification*, qui consiste à mettre en œuvre des techniques d'ordonnancement. Intégrée dans le processus global de conduite, la planification propose une affectation pour les différentes opérations, dans le temps et l'espace. C'est un moyen unique et incontournable pour s'assurer du respect des objectifs fixés. Néanmoins les méthodes de calcul employées pour sa mise en œuvre sont souvent pénalisantes dans le temps.

► *le lancement*, qui répartit et transmet les ordres au système en tenant compte de l'état des entités de production ;

- ▶ *le suivi*, qui recueille l'ensemble des événements survenant dans l'atelier et qui met à jour une image interne du système opérant ;
- ▶ *la réaction*, qui corrige les déviations induites par les aléas de production. Cette activité est très liée à celle de la planification, car il s'agit de prendre des mesures correctives tout en s'assurant du respect des objectifs de production.

La conduite en temps réel d'un système de production est une tâche complexe qui requiert des connaissances dans les domaines de l'informatique, de l'automatique, de la production, de la communication homme - machine ...etc. Sa difficulté est due (Ghazi, 2007):

- à la complexité du problème d'ordonnancement ;
- à la nécessité de résoudre le problème d'ordonnancement dans un contexte perturbé ;
- au problème intrinsèque du suivi, souvent banalisé dans la plupart des systèmes ;
- à l'intégration du système de conduite dans le système global de gestion de production.

Pour gérer le déroulement des opérations sur les moyens de production la conduite doit respecter les contraintes techniques suivantes :

- une machine exécute au plus une opération à la fois (contrainte d'exclusion mutuelle) ;
- une opération commencée sur une machine doit être terminée sans interruption sur la même machine (contrainte de non - préemption) ;
- la fabrication d'une pièce nécessite un ensemble d'opérations qui doivent être effectuées dans un ordre spécifique (contrainte de précédence).

Finalement, pour garantir de meilleures performances d'un système de production, il faut bien le gérer. Pour mieux le gérer, il faut bien le piloter. Pour bien le piloter, il faut s'assurer de sa bonne conduite, l'un des paramètres de la bonne conduite d'un système de production est la résolution du problème d'ordonnancement. C'est le niveau le plus bas de la fonction planification; il consiste à proposer une affectation pour les différentes opérations, dans le temps et l'espace. C'est un moyen unique et incontournable pour s'assurer du respect des objectifs fixés.

III.4. Les problèmes d'ordonnancement :

III.4.1. Définition :

On rencontre les problèmes d'ordonnancement dans de très nombreux types de systèmes dès lors qu'on est amené à organiser des tâches à réaliser, à répartir des ressources, à planifier l'exécution des différentes parties élémentaires d'un projet complexe. Depuis une soixantaine d'années, les études traitant des problèmes d'ordonnancement ont été très nombreuses, que ce soit les problèmes académiques ou bien des problèmes liés à des applications dans le domaine de la production, dans les systèmes informatiques, dans la gestion des emplois du temps ou encore la planification des ressources humaines.

La littérature est maintenant si dense qu'on trouve de nombreuses définitions du terme ordonnancement. Carlier et Chrétienne donnent, dans (Carlier et Chrétienne, 1988), la définition suivante :

Ordonnancer c'est programmer l'exécution d'une réalisation en attribuant des ressources aux tâches et en fixant leurs dates d'exécution.

Plusieurs autres définitions peuvent être citées :

Le problème d'ordonnancement consiste à organiser dans le temps la réalisation de tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement, ...) et de contraintes portant sur l'utilisation et la disponibilité de ressources requises (Ghazi, 2007)

Ordonnancer un ensemble de tâches, c'est programmer leur exécution en leur allouant les ressources requises et en fixant leurs dates de début (GHOTA, 1993).

Ordonnancer la production, c'est planifier les dates de lancement et de fin des opérations ainsi que l'affectation de ces opérations aux différents postes de production susceptible de les exécuter. On classe généralement la fonction d'ordonnancement parmi les activités de planification à très court terme, quoique dans le cas d'ordonnancement de projets elle puisse parfois tenir compte d'un horizon de plusieurs années (Dupont et al, 1997).

L'ordonnancement est la programmation dans le temps de l'exécution d'une série de tâches (ou activités, opérations) sur un ensemble de ressources physiques (humaines et techniques), en cherchant à optimiser certains critères, financiers ou technologiques, et en respectant les contraintes de fabrication et d'organisation (GHOTA, 1993) et (Esquirol et Lopez, 1999). Les ordres de fabrication (OF), suggérés par le calcul des besoins, représentent chacun une requête pour fabriquer une quantité déterminée de pièces pour une date donnée. Ils constituent les données d'entrée de l'ordonnancement et permettent de définir, au moyen des gammes de fabrication, l'ensemble des tâches que la fonction ordonnancement doit planifier.

III.4.2. La problématique d'ordonnancement :

Résoudre un problème d'ordonnancement consiste à trouver une planification des tâches sur les ressources en optimisant les objectifs et en respectant les contraintes. La modélisation du problème dépend fortement de différents paramètres, contraintes, objectifs... Les problèmes d'ordonnancement sont différenciés suivant le nombre de machines (problèmes à machine unique ou multi-machines) et, si l'on a plusieurs machines, suivant l'ordre de réalisation des tâches (GHOTA, 1993) et (Esquirol et Lopez, 1999).

Un ordre de passage des tâches constitue une solution aux problèmes d'ordonnancement. Il décrit l'exécution des tâches et l'allocation des ressources au cours du temps, et vise à satisfaire un ou plusieurs objectifs. Plus précisément, on parle de problèmes d'ordonnancement lorsqu'on doit déterminer les dates de début et les dates de fin des tâches, alors qu'on réserve le terme de séquençement au cas où l'on cherche seulement à fixer un

ordre relatif entre les tâches qui peuvent être en conflit pour l'utilisation des ressources. Un ordonnancement induit nécessairement un ensemble unique de relation de séquençement.

III.4.3. Les critères de performance et objectifs d'un problème d'ordonnancement :

Les objectifs les plus recherchés dans la résolution des problèmes d'ordonnancement sont l'utilisation efficace des ressources, le délai global, la minimisation des encours et le respect des dates échues. Les facteurs usuels permettant de les atteindre sont

Pour la minimisation des encours de fabrication :

$$F_{\max}, \bar{F} \text{ et } \bar{F}_w$$

Pour minimiser le mécontentement des clients

$$T_{\max}, \bar{T} \text{ et } \bar{T}_w$$

Pour minimiser la durée totale de fabrication ou de service :

$$C_{\max} \text{ (Le critère le plus utilisé.)}$$

Les variables intervenant le plus souvent dans l'expression de la fonction économique (le critère à optimiser, l'objectif à atteindre) sont :

La date C_i de la fin d'exécution de la tâche i ;

Le retard $T_i = \max(0, C_i - d_i)$ de la tâche i ;

L'indicateur de retard U_i ($U_i = 0$ si $C_i \leq d_i$; $U_i = 1$ sinon)

La relation entre les différents critères :

Nous nous intéressons dans cette partie aux diverses relations d'équivalence et de similitude qui peuvent exister entre les critères d'optimisation. En effet, une solution optimale pour l'un ou l'autre des problèmes pour un critère donné peut s'avérer optimal pour un autre critère. La figure ci-dessous donne les liens entre certains critères, nous lisons au début, qu'une solution optimale pour le critère L_{\max} est aussi optimale pour le critère C_{\max} (avec $d_i = 0$). Autrement dit, une méthode qui optimise L_{\max} optimise C_{\max} , la méthode qui optimise le critère $\sum W_i C_i$ optimise aussi le critère $\sum C_i$, etc., (Pinedo, 1995) et (Esquirol et Lopez, 1999).

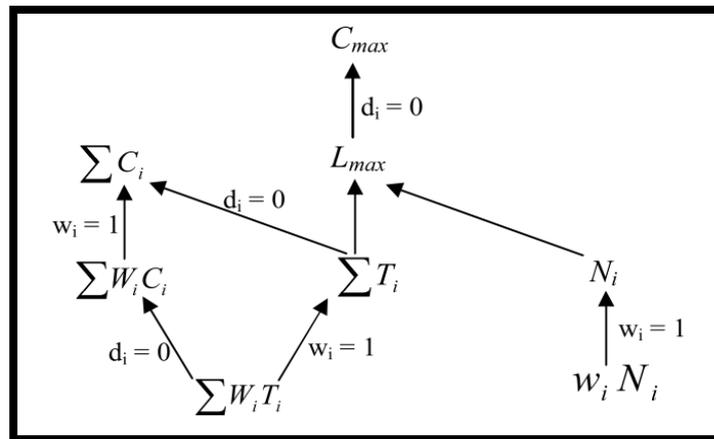


Figure I.10: Les relations entre les différents critères

Les méthodes de résolution usuelles considèrent un seul critère à la fois, or les problèmes industriels sont plus complexes. Un responsable d'atelier est souvent intéressé par un ensemble d'objectifs : minimiser le temps total d'exécution de toutes les tâches, tout en minimisant l'oisiveté des ressources coûteuses. De ce fait, les problèmes d'ordonnancement industriels sont considérés comme des problèmes multicritères. Ces problèmes sont abordés au moyen de la notion de Pareto-dominance. Ainsi, le décideur pourra choisir entre un ensemble de solutions dominantes (front de Pareto) (Behiri et Latreche, 2009).

III.4.4. Les notions fondamentales d'un problème d'ordonnancement :

► Les tâches :

C'est un travail élémentaire dont la réalisation nécessite un certain nombre d'unités de temps (sa durée) et d'unités de chaque ressource (GOTHA, 1993).

Une tâche (job) correspond à la fabrication d'un objet (ou la fourniture d'une prestation d'un service) (Botta, 1999).

On note en général $I = \{1, 2, \dots, n\}$ l'ensemble des tâches et p_i la durée de la tâche i si cette durée ne dépend pas des ressources qui lui sont allouées. Souvent une tâche i ne peut commencer son exécution avant une *date de disponibilité* notée r_i et doit être achevée avant une *date échue* d_i . Les dates réelles de début et de fin d'exécution de la tâche i sont respectivement t_i et C_i .

Une tâche peut être morcelable ou encore préemptive, dans ce cas, elle peut être exécutée par morceaux (en informatique), sinon elle est non préemptive comme c'est le cas en général dans l'ordonnancement d'atelier manufacturier.

► Les opérations :

Lorsque la réalisation d'une tâche requiert successivement plusieurs ressources, on dira que chaque ressource réalise une opération de la tâche considérée. La gamme d'une tâche décrit la séquence des opérations de cette tâche (Ghazi, 2007).

► Les gammes :

Les gammes de fabrication décrivent la succession des opérations à réaliser pour passer d'une matière au composant ou du produit semi-fini au produit fini, c'est le processus d'élaboration. On trouve dans la littérature non spécialisée, l'appellation de routage de la tâche et dans la littérature anglophone le terme : routing. Il existe plusieurs sortes de gammes, dont on distingue (Hentous, 1999) :

- Les gammes libres où gammes non linéaire (c'est le cas considéré dans notre étude) ;
- Les gammes linéaires ;
- Les gammes techniques ;
- Les gammes mixtes ou semi-linéaires. (*pour plus de détails : voir annexe I*)

► Les ressources :

Une ressource j est un moyen technique ou humain destiné à être requis pour la réalisation d'une tâche et disponible en quantité limitée, sa capacité a_j est supposée constante.

On distingue plusieurs types de ressources (Esquirol et Lopez, 1999) et (Lopez et Roublat, 2001):

- Ressources renouvelables ;
- Ressources consommables ;
- Les ressources doublements contraintes ;
- Les ressources disjonctives ;
- Les ressources cumulatives. (*pour plus de détails : voir annexe I*)

► Les contraintes :

Les contraintes de différents types conditionnent l'admissibilité d'un ordonnancement. Elles sont :

- *Contraintes temporelles :*

Respect d'un début au plus tôt, d'une fin au plus tard, d'une date échue...;

- *Contraintes d'antériorité (potentielles) :*

Respect de la gamme ou de la cohérence technologique (durées minimales ou maximales entre tâches) ;

- *Contraintes de ressources :*

- ▶ Disjonctives : les tâches doivent être effectuées l'une après l'autre ;
- ▶ Cumulatives : les besoins en ressources ne doivent pas dépasser les capacités des ressources.

III.5. Le processus global d'ordonnancement :

En prenant en compte l'environnement dynamique de mise en œuvre de l'ordonnancement, le problème global d'ordonnancement peut se décomposer en deux phases (Esswein, 2003) :

1. **La phase statique hors ligne**, où une solution est produite en se basant sur des hypothèses plus au moins restrictives et en n'ayant qu'une vision inévitablement imparfaite du problème à résoudre ;
2. **La phase dynamique en ligne**, où la solution disponible à la fin de la phase statique est effectivement mise en œuvre dans un environnement dynamique.

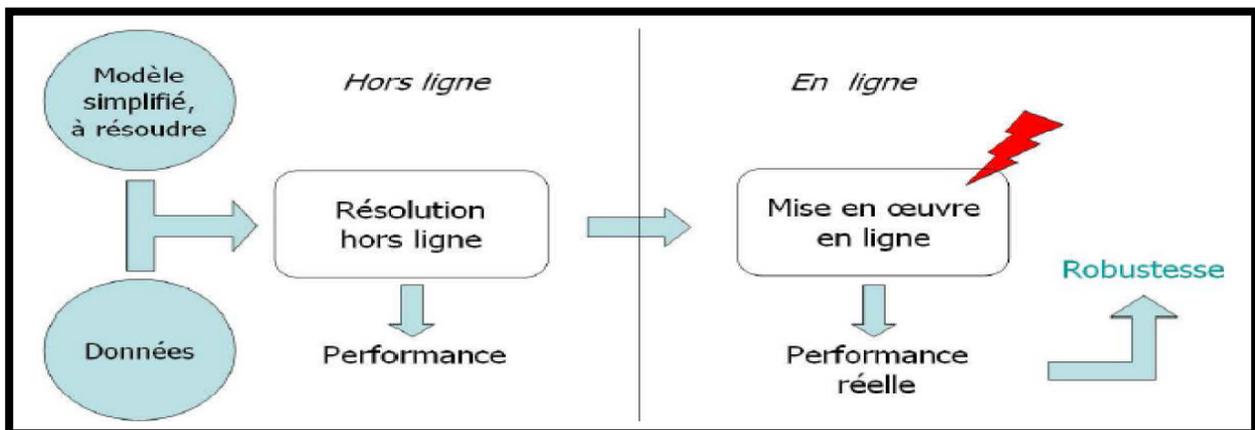


Figure I.11: Processus global d'ordonnancement (Esswein, 2003)

Proposer une méthode d'ordonnancement revient donc à proposer, idéalement, à la fois un ordonnancement statique et un ordonnancement dynamique. Pourtant plusieurs travaux ne portent que sur la phase hors ligne, et négligent l'aspect dynamique de l'ordonnancement (Esswein, 2003). De nos jours nous savons pertinemment, que cette hypothèse est fortement discutable, et de manière générale il faut prendre en compte les incertitudes et les aléas pour répondre de manière efficace à l'aspect indiscutablement dynamique de l'environnement de l'application. Cette prise en compte se fait en développant des méthodes d'ordonnancement robustes (Djellouli, 2007)

III.6. Les organisations d'ateliers : (OS et JS)

Dans les problèmes d'ordonnancement d'atelier, les ressources sont des machines ne pouvant réaliser qu'une tâche (ou opération). Par ailleurs chaque travail concerne une entité physique indivisible appelé produit, ou lot lorsque plusieurs produits identiques sont groupés.

Une entité ne peut pas se retrouver simultanément en deux lieux distincts, un même travail ne peut être exécuté qu'à raison d'une opération à la fois, sur une seule des machines. On distingue les problèmes à machine unique et les problèmes multi-machines. Dans le premier cas, les travaux se réduisent à une seule opération, il faut alors trouver une séquence des travaux sur la machine. Dans l'autre, on distingue tout d'abord les problèmes à machines parallèles, qui conserve l'hypothèse des travaux à opération unique et pour lesquels la solution est un ensemble de séquences, une par machine. Ensuite nous avons les problèmes d'ateliers multi-machines où les travaux comportent plusieurs opérations, chacune nécessitant une machine particulière disponible en exemplaire unique. Ce cas regroupe lui même trois types de problèmes, si l'ordre des opérations composant un travail est fixé et commun à tous les travaux, c'est un atelier à cheminement unique ou Flow Shop. Si l'ordre est fixé mais propre à chaque travail, c'est un atelier à cheminement multiple ou Job Shop. Si l'ordre est indéterminé, alors l'atelier est à cheminement libre ou Open Shop.

On peut regrouper les organisations en trois grandes classes comme suit :

- Les organisations à ressource unique.
- Les organisations à ressources multiples.
- Les organisations multi-lignes parallèles.

La figure I.12 illustre schématiquement les différences et les ressemblances qui existent entre les différentes organisations de production. Un arc reliant une organisation O_1 à une autre organisation O_2 peut être interprété comme suit : O_2 est un cas particulier de la première organisation O_1 .

Ces relations entre organisations illustrent bien les similitudes des systèmes de production.

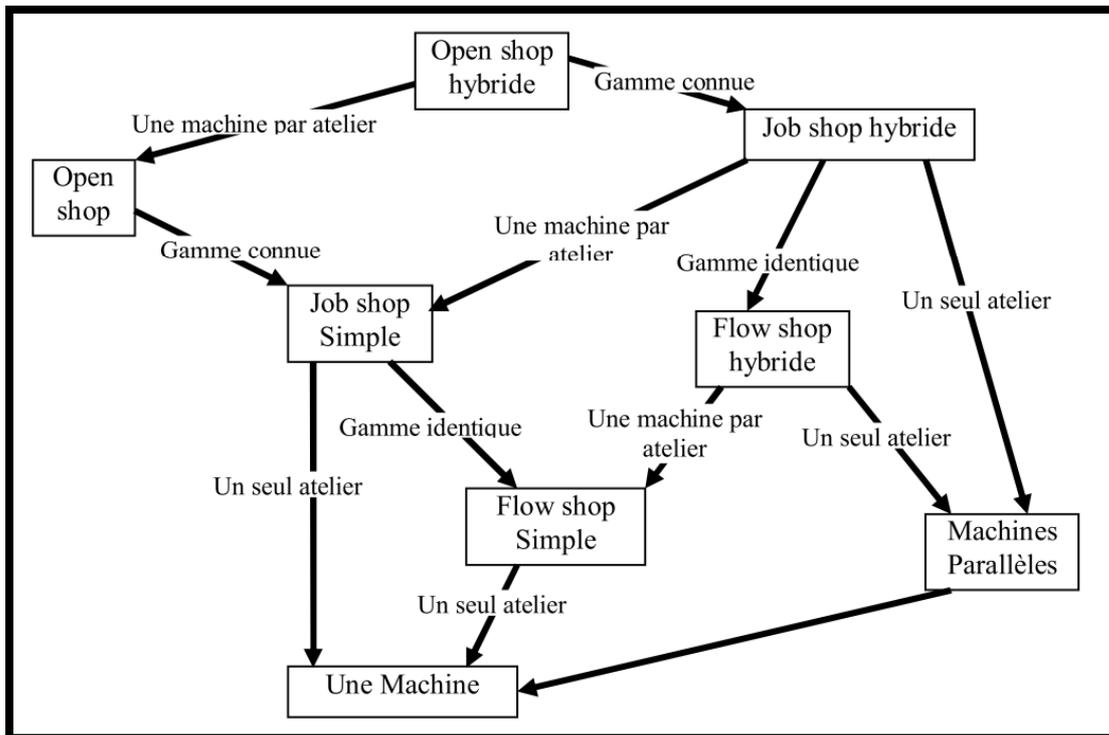


Figure I.12: Schéma montrant les liens entre diverses organisations (Ghazi, 2007)

III.6.1. Les ateliers OS et JS :

Les problèmes d'OS et de JS appartiennent à la classe des problèmes dits d'atelier comme le FS. On les retrouve fréquemment dans les problèmes d'ordonnancement industriels.

Ces problèmes sont NP-Difficiles dans le cas général. Les applications d'OS et de JS sont nombreuses. Historiquement, la structure des ateliers (lignes d'assemblage) a imposé des contraintes de précédence dues au flux des produits entre les machines, ce qui a favorisé les recherches sur le Job-Shop (ainsi que le FS). En fait, avec le développement des cellules flexibles (le cas étudié dans notre travail), de nombreuses contraintes de précédence disparaissent d'une part parce que le produit n'est plus déplacé d'un poste à l'autre suivant le positionnement des postes de travail, et d'autre part parce que des opérations deviennent exécutables sans ordre précis, voire en parallèle. Ainsi, les problèmes d'OS existent bien, et ils ont même tendance à être de plus en plus répandus.

III.6.2. Définition :

Dans les problèmes d'OS et de JS généraux on considère un ensemble M de m machines $\{M_1, M_2, \dots, M_m\}$ et un ensemble N de n jobs ou travaux $\{J_1, J_2, \dots, J_n\}$ à exécuter sur ces machines. Chaque job J_j est composé de m tâches ou opérations $O_{1j}, O_{2j}, \dots, O_{mj}$. Chaque tâche O_{ij} a une durée opératoire p_{ij} et doit être exécutée sur la machine M_i . Une instance du problème est donc définie par deux entiers m et n , et par une matrice $P(m \times n)$ des temps opératoires. Les contraintes du problème sont les suivantes :

- une machine ne peut exécuter qu'une seule tâche à la fois.
- plusieurs tâches d'un même job ne peuvent être exécutées en même temps.
- pour le JS le séquençement des jobs est imposé.
- contrairement aux JS, le séquençement dans le cas de l'OS des jobs n'est pas imposé.

III.6.3. Quelques méthodes de résolution existantes :

a). Heuristiques de liste classiques :

Voici un aperçu des heuristiques de liste classiques utilisées pour résoudre les problèmes d'OS et de JS. Ces heuristiques classent les h tâches de durées non nulles selon un ordre de priorité. Les tâches sont ordonnancées à partir de l'instant de départ t_0 en affectant à l'instant t , parmi les tâches disponibles, celle de plus grande priorité. Les tâches disponibles à un instant donné sont toutes les tâches O_{ij} non encore ordonnancées et telles qu'aucune tâche du job J_j et aucune tâche de la machine M_i ne soit en cours d'exécution à cet instant.

Il s'agit d'algorithmes construisant des ordonnancements sans retard* (no-delay schedules). Ces algorithmes peuvent ne pas trouver l'optimum, quelle que soit la liste de priorités. On peut les adapter pour construire des ordonnancements actifs, qui incluent l'optimum, mais les performances moyennes sont alors bien moins bonnes.

L'algorithme 1 (figure I.13) présente la structure générale de telles heuristiques : I est l'ensemble des tâches à ordonnancer, U l'ensemble des tâches ordonnancées et $D(t)$ l'ensemble des tâches de $I - U$ disponibles à l'instant t .

```

U := ∅
Tant que U ≠ I
    Déterminer le premier instant  $t_{\min}$  où  $D(t_{\min})$  n'est pas vide
    Choisir la tâche  $O_{ij}$  de  $D(t_{\min})$  de plus grande priorité
     $t_{ij} := t_{\min}$ 
     $U = U \cup \{O_{ij}\}$ 
Fin Tant que

```

Figure I.13: Algorithme 1 (Structure générale d'un algorithme de liste (cas sans retard) (Ghazi, 2007)

Les heuristiques de liste les plus connues sont la LPT (Longest Processing Time) et la SPT (Shortest Processing Time). Il s'agit d'idées simples et naturelles d'algorithmes, déjà utilisées pour les autres problèmes d'atelier. Ces heuristiques ont été évaluées sur divers types d'instances générées aléatoirement par (Gueret, 1997).

b). LPT (Longest Processing Time):

L'heuristique LPT donne la priorité aux tâches de plus longues durées. L'algorithme associé est présenté dans la figure I.14. Dans cet algorithme, deux tableaux sont utilisés EOM

(comme End Of Machine) et EOJ (comme End Of Job) dans lesquels nous stockons respectivement les dates de disponibilité des machines et des jobs (Gueret, 1997).

```

U := ∅
EOM:=0
EOJ:=0
Tant que U ≠ I
  (* Déterminer le premier instant tmin où D(tmin) n'est pas vide *)
  Pour chaque tâche Oij libre faire
    t := max(EOM [i]; EOJ [j])
    Si t < tmin alors
      tmin := t
      D(tmin) := {Oij}
    FinSi
    Si t = tmin alors D(tmin) := D(tmin) [ fO ij g
  FinPour
  Choisir la tâche Oij de D(tmin) de plus grande priorité
  tij := tmin
  EOM [i] := tmin + pij
  EOJ[j] := tmin + pij
  U = U ∪ {Oij}
Fin Tant que

```

Figure I.14: Algorithme 2 (Algorithme de liste LPT) (Gueret, 1997)

c). **SPT (Shortest Processing Time):**

Dans l'heuristique SPT, à chaque itération, la tâche sélectionnée est la tâche libre de durée minimale. Cette heuristique a la même complexité que LPT.

Les heuristiques utilisées dans notre étude sont les suivantes :

- **LPT** ;
- **SPT** ;
- **FIFO (First In First Out)** : qui donne la priorité à l'opération appartenant au travail arrivant le premier à la file d'attente ;
- **MWKR (Most Work Remaining first)** : qui donne la priorité à l'opération appartenant au travail pour le quel le reste à faire est le plus grand ;
- **LWKR (Least Work Remaining first)** : qui donne la priorité à l'opération appartenant au travail pour le quel le reste à faire est le plus petit.

d). **LAPT (Longest Alternate Processing Time first):**

Dans la règle LAPT, lorsqu'une machine est libérée, on sélectionne parmi les travaux en attente celui dont la durée opératoire sur l'autre machine est la plus grande.

III.6.4. La complexité des problèmes d'ordonnancement OS et JS

Dans cette section nous aborderons la complexité des problèmes OS et JS. C'est à ce niveau que nous décidons de l'approche que nous adoptons pour la suite de notre travail.

La théorie de la complexité a été créée dans le but de fournir des informations sur la difficulté théorique des problèmes que l'on peut être amené à résoudre. Concrètement, on cherche à déterminer " mathématiquement ", si le problème étudié est plutôt " facile " ou bien s'il est plutôt " difficile ". Des classes de problèmes ont été définies, et certaines propriétés qui sont associées à ces classes aident à répondre à de telles questions. Pratiquement la connaissance de telles propriétés permet souvent de guider dans le choix de l'algorithme à développer (Esswein, 2003).

Dans la théorie de la complexité, la difficulté d'un problème est mesurée par la complexité du meilleur programme résolvant ce problème de façon optimale.

Un algorithme est dit polynomial si son temps d'exécution est borné par une fonction polynomiale de la taille de l'instance. On dit qu'il a une complexité en $O(p(n))$, où p est un polynôme et n la taille de l'instance d'entrée. Dans le cas contraire, il est dit exponentiel.

Pour des problèmes d'optimisation, on parle d'un problème P lorsqu'il existe un algorithme polynomial permettant de le résoudre. Un problème est dit NP lorsqu'il existe un algorithme non déterministe polynomial pour le résoudre. Un problème est NP -difficile lorsqu'aucun algorithme de complexité $O(p(n))$, ne peut le résoudre. Cette classe de problèmes est divisée en deux :

- Les problèmes NP -difficiles au sens faible, que l'on peut résoudre avec un algorithme de résolution en $O(p(n,l))$, où p est un polynôme, n la taille de l'instance d'entrée, et l le plus grand nombre de l'instance ;
- Les problèmes NP -difficile au sens fort, qu'aucun algorithme en $O(p(n,l))$ peut résoudre.

La plupart des problèmes d'ordonnancement sont des problèmes NP -difficile au sens fort. Le problème Job Shop est classé dans la théorie de la complexité parmi les problèmes NP -difficile au sens fort. Nous pouvons trouver la démonstration de ce résultat dans (Lenstra et Rinnoy Kan, 1978).

De nombreux travaux ont été effectués sur la complexité des problèmes d'Open-Shop. Très peu de problèmes Open Shop peuvent être résolus en temps polynomial. Les cas les plus connus sont $O2//C_{\max}$ et $O/pmtn/C_{\max}$ (Gonzales et Sahni, 1976).

Pour tous les critères usuels, les problèmes Open-Shop sans attente ont été prouvés NP -Difficiles au sens fort par (Sahni et Cho, 1979) même pour les problèmes ne comportant que deux machines.

Le problème $O2/\tau_j/C_{\max}$ est NP-difficile au sens fort (Dell'Amico et Vaessens, 1995) Le problème $O2/\tau_j/C_{\max}$ avec des temps égaux de transport est connu pour être NP-difficile binaire tel prouvé dans (Rayward-Smith et Rebaïne, 1992). Un exemple de problèmes à traiter dans ce travail est $O4/\tau_j/C_{\max}$.

En résumé, le problème d'Open-Shop non-préemptif est NP-Difficile dans le cas général. Dans notre cas il est NP-difficile au sens fort.

IV. Conclusion

Dans ce chapitre, on a abordé toutes les notions élémentaires essentielles à notre étude afin de l'entamer sans aucune ambiguïté.

A cet effet, on a d'abord introduit la notion de production, les systèmes de production et la fonction d'ordonnancement sachant que l'objectif de notre étude est l'optimisation du fonctionnement d'un atelier cyclique d'assemblage. Le système étudié est un SFM c'est-à-dire une cellule de production flexible.

Comme on l'a déjà souligné, dans l'étude des problèmes OS et de JS, ces derniers sont NP-difficile dans le cas général. Même si la préemption est autorisée, ces problèmes sont déjà NP-difficile. Or, dans notre cas, la préemption est non autorisée (ressources disjonctives), la typologie est circulaire (ce qui rend les problèmes plus combinatoire par rapport aux lignes d'assemblage conventionnelle) et les temps de transport sont considérés (considération des ressources de transport) ce qui fait de nos problèmes des problèmes NP-difficile au sens fort. Les problèmes considérés sont $O_n/\tau_j/C_{\max}$ et $J_n/\tau_j/C_{\max}$ où $n \in \{4,5,6,7,8,9,10\}$.

Vu la complexité de ces types de problèmes, on a opté pour une approche empirique de prise en charge du problème par utilisation de la simulation informatique. Ce choix est dû à sa puissance comme outils d'analyse des systèmes complexes et la résolution des problèmes d'optimisation des flux physiques ou des flux d'informations, dans les systèmes de production manufacturiers. Ainsi les trois prochains chapitres sont consacrés à l'étude et la modélisation des problèmes énoncés ci-dessus, c'est-à-dire à la partie pratique de ce mémoire.

Chapitre II

MODELISATION DU SYSTEME D'ASSEMBLAGE

*“La science consiste à passer d'un
étonnement à un autre”*

[Aristote]

Chapitre II : Modélisation du système d'assemblage

- I. Introduction
- II. La modélisation du système étudié
 - a). Les modélisations OS et JS
 - b). Les modélisations FS et l'approche collaborative
- III. La description du système cyclique d'assemblage
- IV. L'analyse du système d'assemblage
- V. La simulation des systèmes de production
 - V.1. La modélisation pour la simulation des systèmes de production
 - V.2. Les définitions de la simulation
 - V.3. Les domaines d'application de la simulation
 - V.4. Les étapes des études de simulation
 - V.5. Les avantages et inconvénients de la simulation
- VI. Présentation du logiciel ARENA 12
 - VI.1. Exemple d'un modèle (de simulation) ARENA
 - VI.2. L'application VBA de Microsoft sous ARENA
 - VI.3. Illustration (model 3- 1)
- VII. Conclusion.

I. Introduction

Dans le cadre de ce travail, on a retenu la simulation comme support principal de mise en œuvre de l'approche empirique adoptée pour la résolution du problème posé. A cet effet, on tentera de répondre à certaines questions en vue de vérifier la possibilité d'améliorer les performances du système étudié.

Dans le souci de garder un enchaînement logique à ce chapitre, et avant d'aborder l'essentiel du sujet (élaboration du modèle de simulation), on s'intéressera d'abord à la description de la cellule d'assemblage ainsi que la justification des approches de résolution retenues. Dans la deuxième section on abordera la définition et le concept de la simulation, son application au domaine de la production, ses objectifs, apports et limites, la modélisation et les étapes d'un processus de simulation (telles que présentées par (Pritsker, 1986)). La dernière section est consacrée à la présentation du logiciel utilisé.

II. La modélisation du système étudié

Dans un premier temps, on commencera par rappeler la nature complexe des problèmes d'ordonnancement étudiés, on justifiera les approches de modélisation adoptées, puis on présentera une description détaillée de la cellule d'assemblage concernée par ces types d'ordonnancement.

a). Les modélisations OS et JS

Dans cette étude, on étudie le comportement d'une cellule flexible d'assemblage dont la typologie est circulaire, et dont les problèmes d'ordonnancement considérés sont du type Open-shop et Job-Shop (ainsi que Le FS et l'approche collaborative). Sachant que ces types de problèmes d'ordonnancement sont déjà NP-difficile dans le cas général, la considération des ressources de transfert et la préemption non autorisée le rend encore plus difficile. De plus, la typologie circulaire de la cellule rend le problème plus combinatoire par rapport aux lignes d'assemblage conventionnelles.

Et encore plus :

- les pièces nécessaires à l'assemblage d'un produit fini sont contenues sur des palettes, dont le nombre est limité ;
- les pièces sont stockées dans une file d'attente à l'entrée du convoyeur jusqu'à ce que l'une des palettes du convoyeur central se libère et soit réquisitionnée pour transporter cette pièce ;
- les pièces ne sont pas toutes disponibles au même moment ($ri \neq 0$), les différents types de pièces arrivent progressivement à des intervalles réguliers.

Ainsi, ces problèmes sont NP-difficiles au sens fort. Dans le but de prendre en considération tous les paramètres de variation de cette étude sans s'imposer aucune restriction du problème, on a opté pour une approche empirique de prise en charge de ce problème dont la simulation sera le support principal.

Notre objectif est la réduction du temps total de séjour des pièces (C_{max}) dans le système cyclique d'assemblage dont une description détaillée sera développée ci-après.

b). Les modélisations FS et l'approche collaborative :

On a aussi opté pour une approche empirique de prise en charge du problème Flow-Shop et de l'approche collaborative dont la simulation sera aussi le support principal.

Le Flow-Shop (les performances du FS) nous sert dans notre étude comme référence. En effet, On comparera l'impact des différents ordonnancements (OS et JS) sur les performances du système étudié par rapport aux performances du FS du même système.

Concernant l'approche collaborative, on a déduit ci-dessus que le problème d'ordonnement OS considéré est NP-difficile au sens fort. On optera donc pour un ordonnancement temps réel où le routage des opérations d'un travail est décidé en temps réel.

Pour plus de flexibilité et de réactivité, on dotera notre système d'une architecture distribuée inspirée des Systèmes Multi-Agents (SMA) pour une optimisation du compromis attente-transfert.

Ce système permettra de prendre en compte simultanément l'état des différentes ressources pour déterminer un meilleur routage de la pièce à travers le système afin de compléter son assemblage. Au fur et à mesure, la prochaine destination de la palette n'est plus la station la plus proche disposant de place dans sa file d'attente d'entrée mais celle dont la file d'attente est le **moins saturée** et la **plus proche** sur le convoyeur. Celle-ci sera déterminée à l'issue du **processus de coopération** entre les différentes ressources ou centres de décisions (voir chapitre III pour plus de détails).

L'ordonnement des opérations au niveau des files d'attente d'entrée est effectué par application des règles de priorité FIFO.

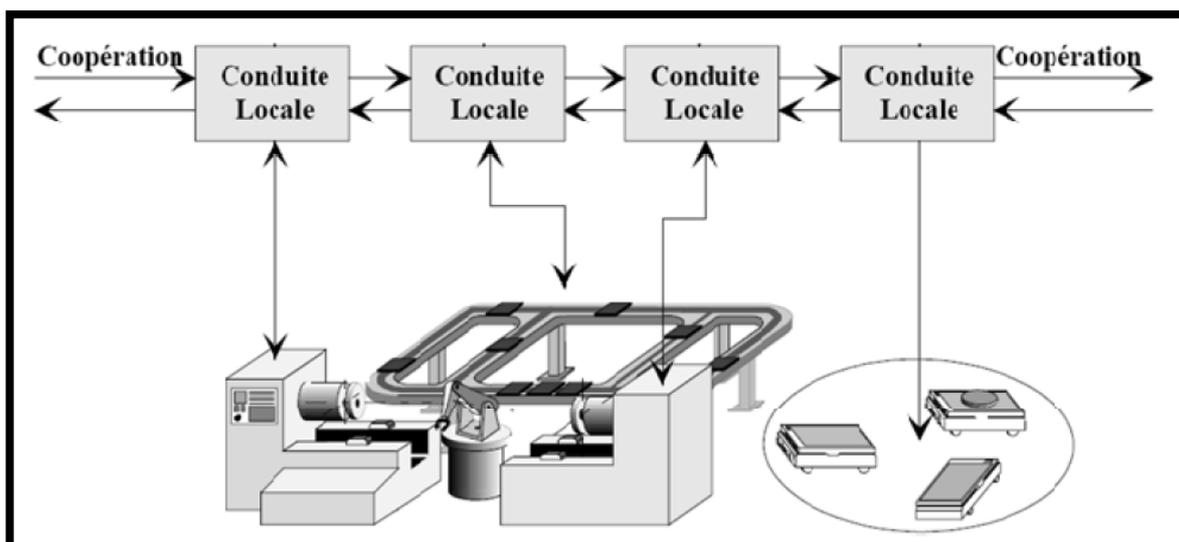


Figure II.1 : Architecture du système d'ordonnement basé sur la coopération
(Baba Aissa et Maouassa, 2008)

III. La description du système cyclique d'assemblage (Un exemple à 4 stations)

La cellule d'assemblage cyclique étudiée (exemple) est détaillée dans la figure II.2, elle est constituée de:

- ▶ Un convoyeur en boucle dont la dimension est choisie par l'expérimentateur.
- ▶ Un nombre fini de palettes dont le nombre est un pourcentage de la taille de commande de pièces à assembler.
- ▶ Les points d'entrée et de sortie du système :
 - entrée du système ;
 - sortie du système ;
 - les entrées des stations ;
 - les sorties des stations.
- ▶ Les stations sont disposées en boucles attachées au convoyeur, chacune d'elles est définie par:
 - Le type de robot qui exécute l'opération élémentaire et son temps d'exécution;
 - Le temps opératoire total (selon les gammes opératoires des produits);
 - Les Positionnement des entrées et sorties des stations : entrée de la station vers la file d'attente, entrée du robot à partir de la file d'attente d'entrée ,sortie du robot vers la file d'attente de sortie, sortie de la station à partir de la file d'attente de sortie.
 - La capacité NFAE des Files d'Attente d'Entrée (FAEs) devant chacun des robots.
 - La capacité NFAS des Files d'Attente de Sortie (FASs) de chacun des robots.
 - Les Points de branchement avec le convoyeur : à l'entrée de chacune des stations (test d'entrée) et à la sortie de chacune des stations (priorité au convoyeur).

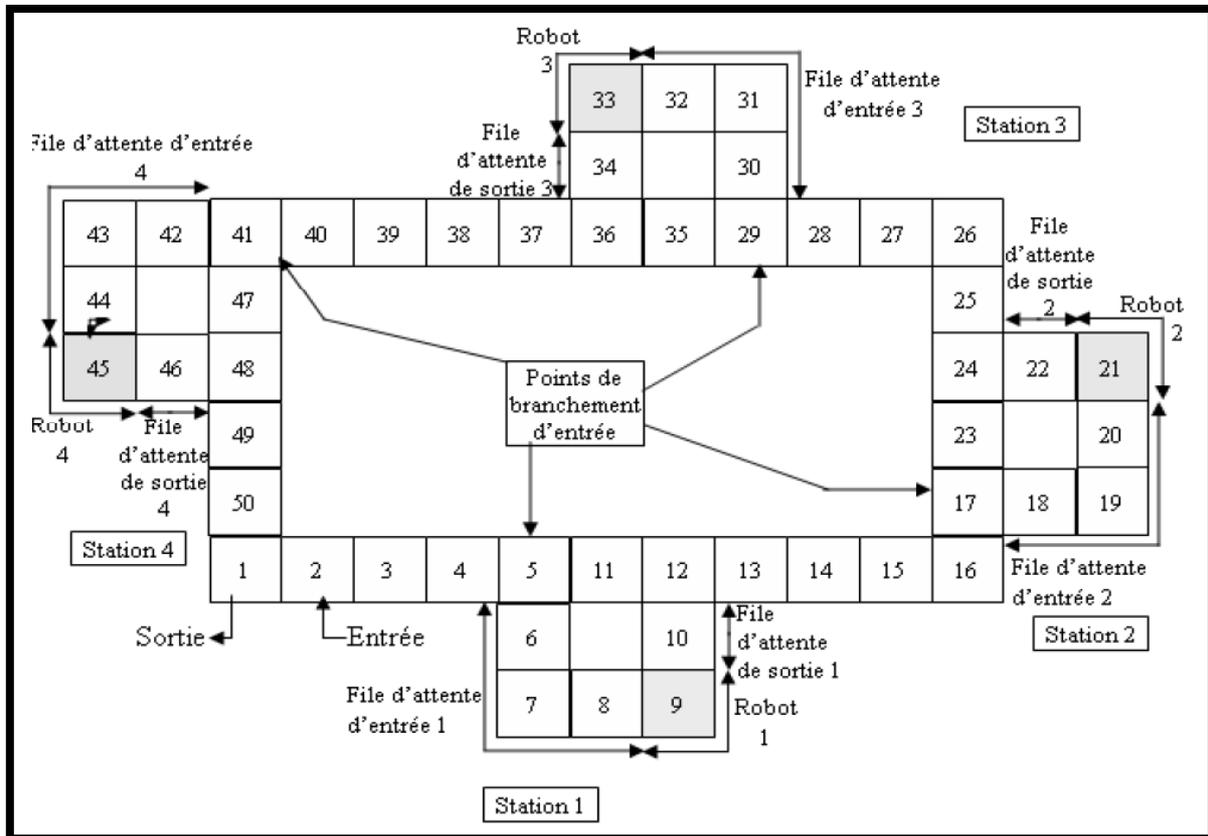


Figure II.2 : Exemple de cellule flexible d'assemblage étudié.

IV. L'Analyse du système d'assemblage

Les stations de travail sont arrangées dans un ordre circulaire autour d'un convoyeur en boucle non bloquant à vitesse régulière. Les pièces à assembler sont toujours montées sur des palettes. Ces palettes sont associées aux pièces à assembler tout au long de leur présence dans le système et elles ne sont déchargées qu'au moment de la sortie des pièces assemblées (produit fini) du système. Ces palettes circulent en permanence sur le convoyeur. Lorsqu'une palette arrive à un point de branchement d'une station, un dispositif de lecture vérifie si la prochaine opération à effectuer est dans la station considérée, si c'est le cas, la palette pénètre dans la file d'attente d'entrée. Dans les autres cas, elle continue à circuler. À la fin de l'assemblage d'un produit fini ou semi-fini, la palette correspondante est déchargée au niveau de la sortie du système et tout de suite réintégrée à l'entrée du système pour une réutilisation.

Le fonctionnement du système selon le cas OS ou JS est décrit dans la figure suivante (le cas du FS est similaire à celui de l'OS avec une capacité infinie des files d'attente):

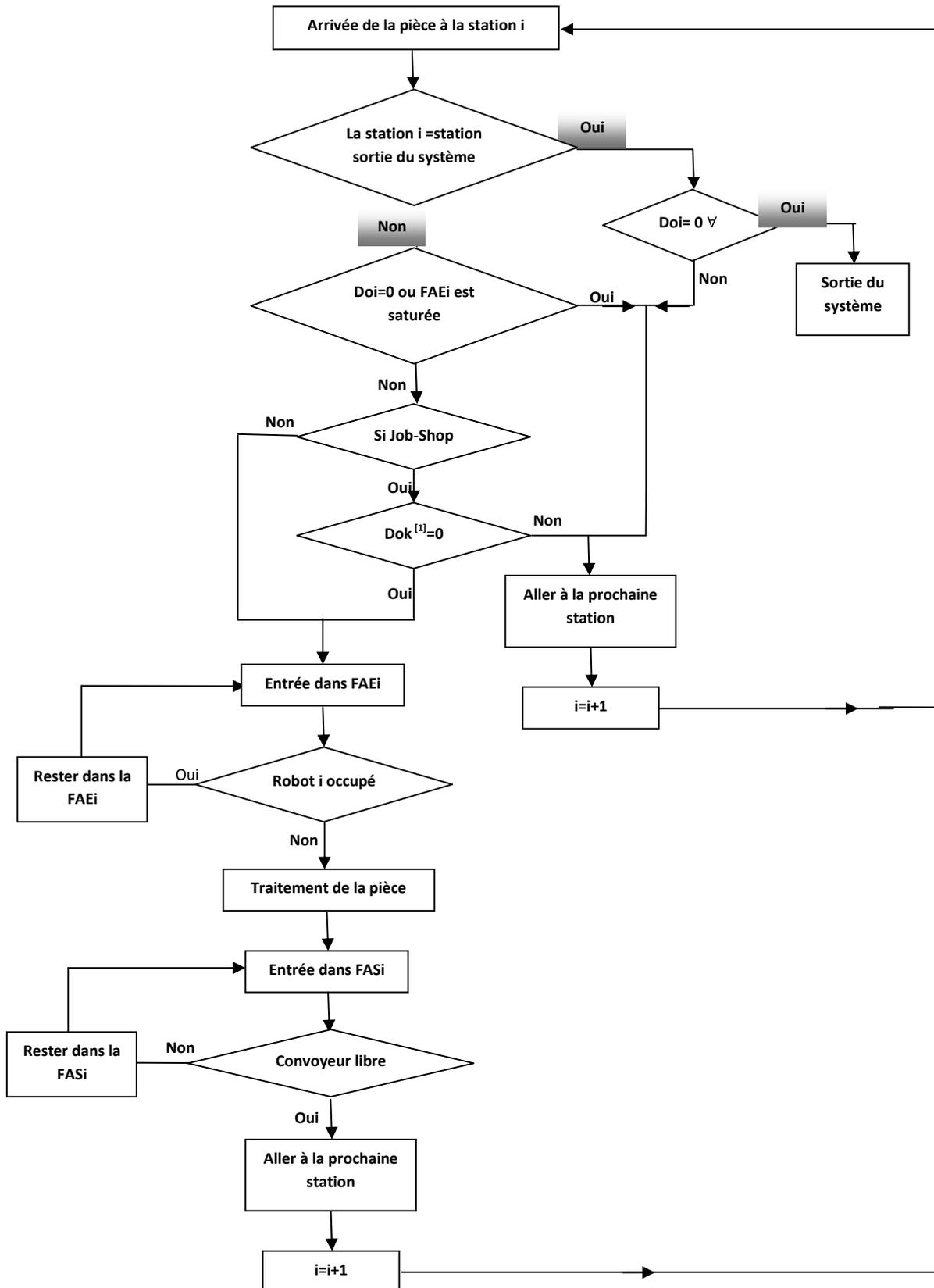


Figure II.3 : Fonctionnement du système selon l'ordonnancement choisi.

[1] : Dok =0 car la station k doit être visitée avant la station i dans le cas Job-Shop.
 Pour les symboles FAEi, Doi,Etc. Se référer à la liste d'abréviation en début de mémoire.

Le fonctionnement du système selon le cas approche collaborative est décrit dans la figure suivante :

Cas 1 : Arrivée d'une pièce ne disposant pas d'engagement (EG=0)

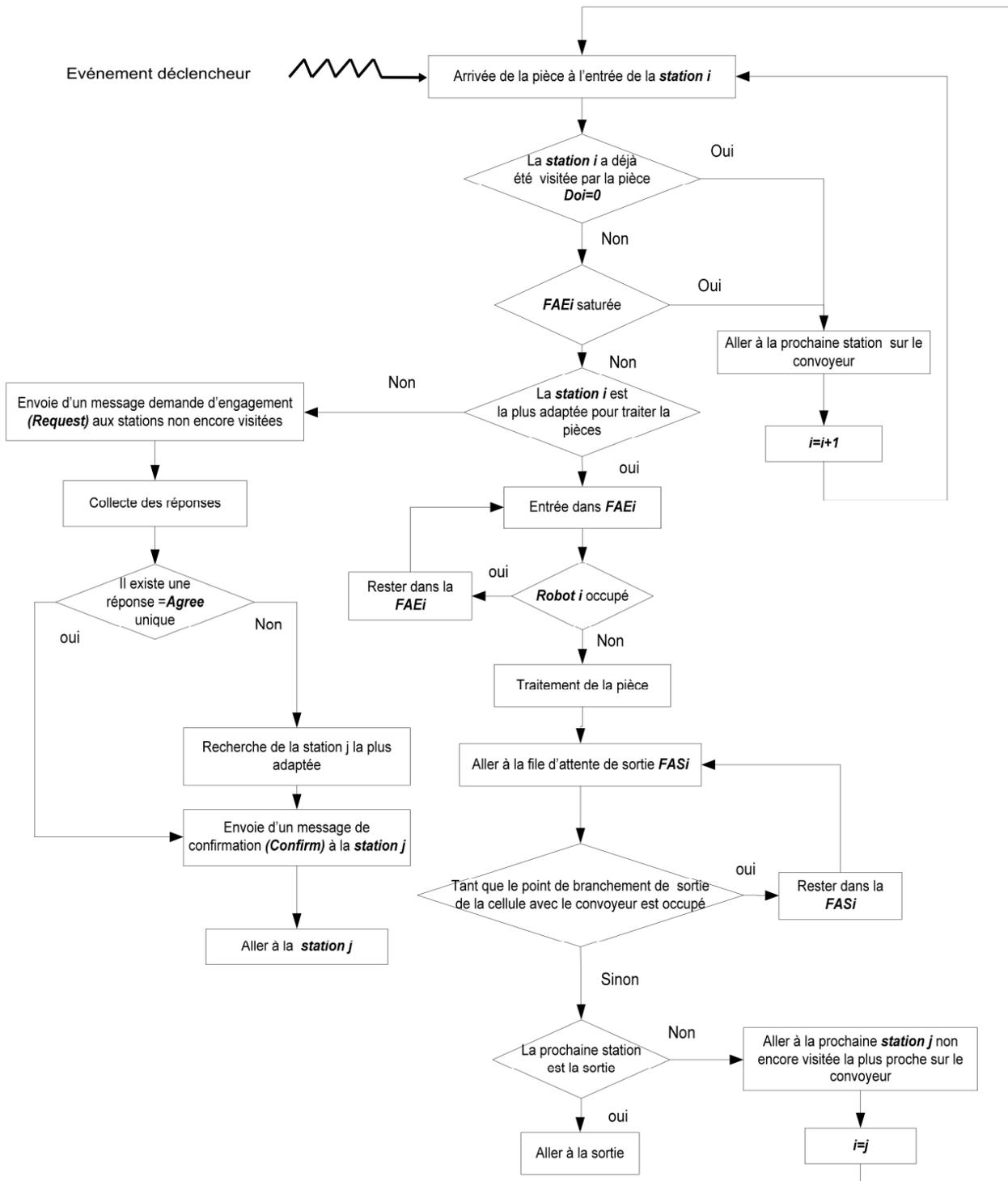


Figure II.4 : Cas 1 : Arrivée d'une pièce ne disposant pas d'engagement (Baba Aissa et Maouassa, 2008).

Cas 2 : Arrivée d'une pièce disposant d'un engagement (EG≠0)

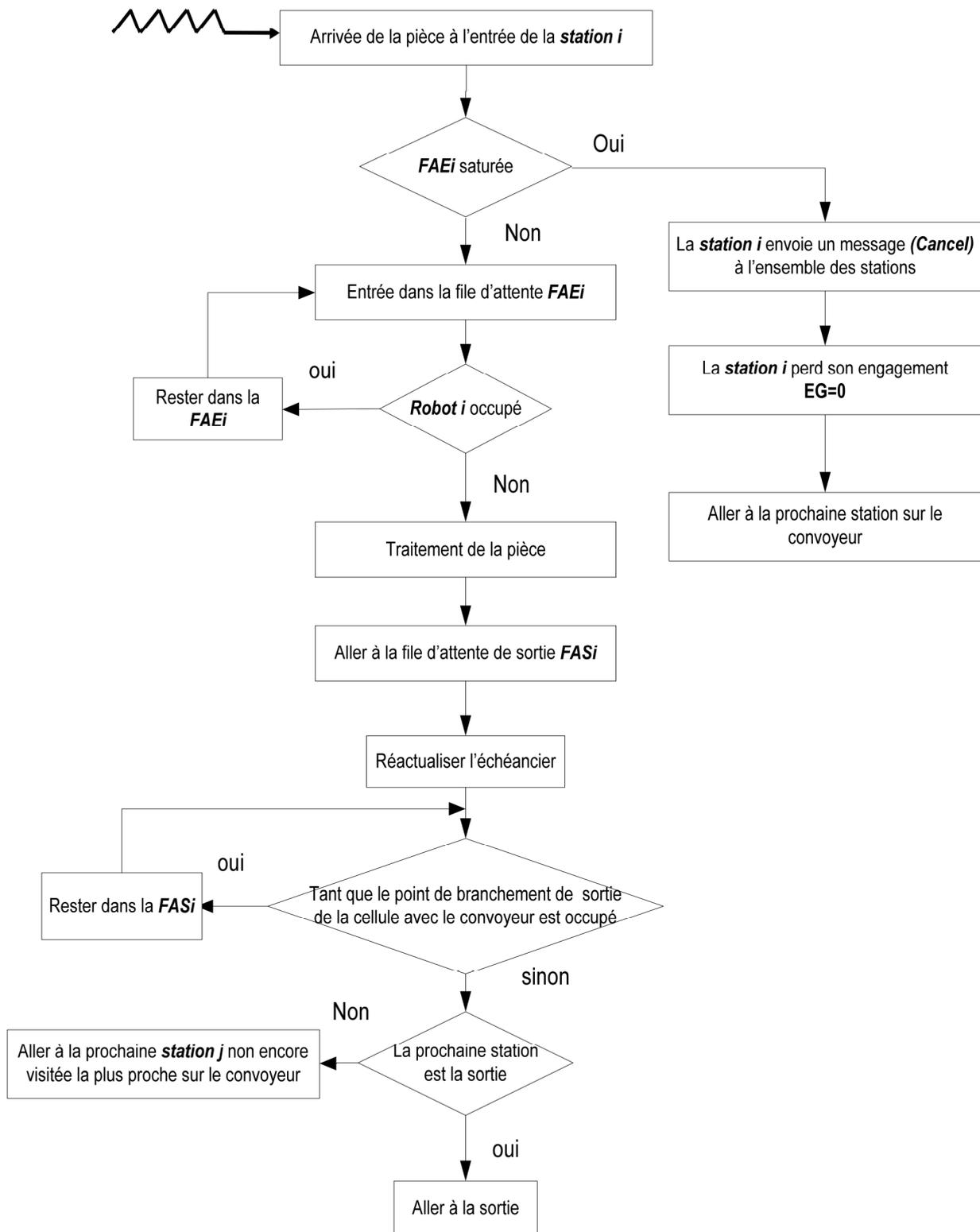


Figure II.5 : Cas 2 : Arrivée d'une pièce disposant d'un engagement (Baba Aissa et Maouassa, 2008).

V. La simulation des systèmes de production

Aujourd'hui indispensable pour l'aide à la prise en charge des problèmes d'optimisation des flux physiques et/ou des flux d'informations, dans les systèmes de production manufacturiers, la simulation est une technique de plus en plus utilisée dans le milieu industriel, c'est l'un des outils qui peut aider à répondre à des questions décisives du type : la gestion appliquée sur cette ligne de production est-elle bien adaptée ? Les paramètres de gestion choisis sont-ils les meilleurs ? Le pilotage appliqué est-il bien adapté ? Comment convaincre le responsable du budget qu'il devient indispensable d'investir dans telle ou telle ressource ? ...et bien d'autres problèmes qu'on peut soulever en milieu industriel :

- L'entreprise doit accroître sa capacité de réponse au marché (réactivité, anticipation...);
- Plus question d'améliorer un indicateur de performance mais de les améliorer conjointement en prenant en compte leur interaction (approche systémique) ;

Les entreprises doivent donner plus d'attention à la manière dont les produits et les services sont fournis et pas uniquement à la nature des produits et des services fournis.

Avant d'aborder les concepts propres de la simulation, la connaissance de ceux de la modélisation s'avère nécessaire.

V.1 La modélisation pour la simulation des systèmes de production

Dans un processus de simulation, la modélisation est l'étape la plus importante dont dépend en grande partie la réussite de ce processus.

Les activités de modélisation consistent à construire un modèle à partir d'un système réel et définir la validité de ce modèle par rapport à ce système (Zeigler, 1976) La simulation quant à elle, va utiliser ce modèle pour analyser le comportement du système réel, dans un environnement donné (Bel et Dubois, 1985). En effet, ce comportement varie selon le contexte d'exploitation (défini par l'environnement) de ce système, ce changement se traduirait par des valeurs de paramètres différentes.

a). Définition de la modélisation :

La modélisation est le processus de traduction d'un système réel, d'abord en un modèle conceptuel, puis en un modèle informatique. La première étape sert à identifier les éléments importants du système réel et les relations qui existent entre eux. La seconde étape quant à elle, sert à les décrire à l'aide d'un langage informatique pour les exécuter.

Ce processus de modélisation est défini par J-L Le Moigne comme "une élaboration et construction intentionnelle, par composition de symboles, de modèles susceptibles de rendre intelligible un phénomène perçu complexe et d'amplifier le raisonnement de l'acteur projetant une intention délibérée au sein de phénomène, raisonnement visant notamment à anticiper les conséquences de ces projets d'actions possibles" (Ghazi, 2007).

Qui dit évaluation dit modèle. On utilise un modèle quand on veut comprendre un système réel que l'on ne peut pas observer ou expérimenter directement, parce que le système n'existe pas encore ou parce que cela est trop difficile à manipuler. Ensuite, par simulation, il est possible de mesurer l'impact relatif de chacune de ces composantes sur la performance globale du système de production.

Ainsi, la modélisation et la simulation de la partie opérative du système permettent de fournir les informations nécessaires au système de conduite (Draghici, 1998).

b). Définition d'un Modèle :

C'est un système physique, mathématique ou logique représentant les structures essentielles d'une réalité, capable à son niveau d'en expliquer ou d'en réduire dynamiquement son fonctionnement.

C'est une abstraction de la réalité pour mettre en valeur le mieux possible les effets et les causes du phénomène, son fonctionnement...etc.

C'est une structure simplifiée de la réalité qui ne contient que les caractéristiques estimées importantes pour l'étude du système en fonction des objectifs que l'on cherche à atteindre.

On utilise un modèle quand on veut comprendre un système réel que l'on ne peut pas observer ou expérimenter directement, parce que le système n'existe pas encore ou parce que cela est trop difficile (coûteux) à manipuler.

Un modèle de simulation permet de prévoir les performances du futur système de production en fonction des choix qui sont faits tant en ce qui concerne le dimensionnement du système physique que de la conception du système de commande. Ce modèle de simulation entre dans une boucle d'optimisation permettant par améliorations successives, d'affiner la conception du système. Le modèle de simulation est structuré en deux parties :

- *Le modèle du système physique* : traduit le comportement de la partie opérative du futur système de production.
- *Le modèle de commande* : traduit le comportement de la partie commande du futur système de production.

Pour bâtir un bon modèle (traduisant au mieux la réalité) l'abstraction construite à partir de la réalité doit capturer ce qui est essentiel pour comprendre le phénomène étudié, et ne pas s'embarasser des éléments qui ne jouent aucun rôle dans ce phénomène. La définition des objectifs de l'étude prend ici toute son importance.

V.2. Définitions de la simulation

La simulation est une technique de modélisation informatique du monde réel (développer un programme informatique qui représente, qui remplace le système réel) pour le comprendre, l'analyser, l'expérimenter, l'améliorer et sans :

- Perturbation de son fonctionnement ;
- Risque de destruction ;
- Investissement important...

La simulation d'un système ou d'un organisme est l'expérimentation d'un modèle qui est une représentation du système ou de l'organisme. Le comportement du modèle peut être étudié et on peut en tirer les propriétés du comportement du système réel ou d'un de ses sous-systèmes.

La simulation implique la génération d'une histoire artificielle du système, et l'observation d'une telle histoire pour connaître les caractéristiques opératoires du système réel.

Au "sens large", la simulation peut être définie comme étant "une science de reproduction du comportement d'un système complexe difficilement contrôlable et soumis à des phénomènes aléatoires ; le comportement du système réel avec toutes ses composantes (physique, informationnelle, décisionnelle) est ainsi remplacé par un modèle virtuel ayant un comportement analogue traduit ensuite en un programme informatique permettant de réaliser des expériences afin d'appréhender le comportement du système réel, d'évaluer ses performances et d'aider à l'anticipation d'éventuelles dérives" (Habchi, 2001).

La simulation est une technique expérimentale appropriée à l'étude des systèmes de grande taille composés de plusieurs éléments en interaction. Elle permet ainsi de répondre à certains problèmes à chaque fois que l'expérimentation en grandeur nature se révèle impossible et/ou trop coûteuse (Carlier et Chrétienne, 1988).

La simulation est principalement utilisée pour étudier les flux dans l'atelier (pièces, matières, outils, informations...) et les disponibilités des ressources (opérateurs, machines, convoyeurs,...). Elle est utilisée en général pour évaluer et comparer des scénarii possibles. Ses capacités d'imitation et de prédiction permettent d'obtenir des renseignements sur les conséquences de changements ou de modifications dans l'atelier (niveau physique ou décisionnel), avant que ceux-ci ne soient effectués (Pegden, 1990).

La simulation est une technique temporelle alors que les techniques non temporelles comportent une série de techniques faisant partie de la recherche opérationnelle (la programmation linéaire, la théorie des files d'attente, la théorie des graphes...etc.). La simulation permet d'évaluer les performances d'un système de configuration donnée aussi bien au niveau de la structure physique (conception de nouveaux systèmes), qu'au niveau du système de décision (exploitation, réorganisation).

La simulation permet de répondre à la question : « *Qu'obtiendra-t-on si l'on fait ceci ?* » mais ne permet pas de répondre à la question : « *Que faut-il faire pour obtenir cela ?* ».

Enfin, la simulation est une démarche par induction (étude de cas particuliers – scénarios – afin d'aboutir à une conclusion, la plus générale possible) et non par déduction (solution obtenue par un raisonnement, un algorithme).

V.3. Les domaines d'application de la simulation

La simulation est utile lorsque le nombre d'éléments d'un système ou le nombre de phénomènes stochastiques est important et lorsque le nombre d'interactions croît.

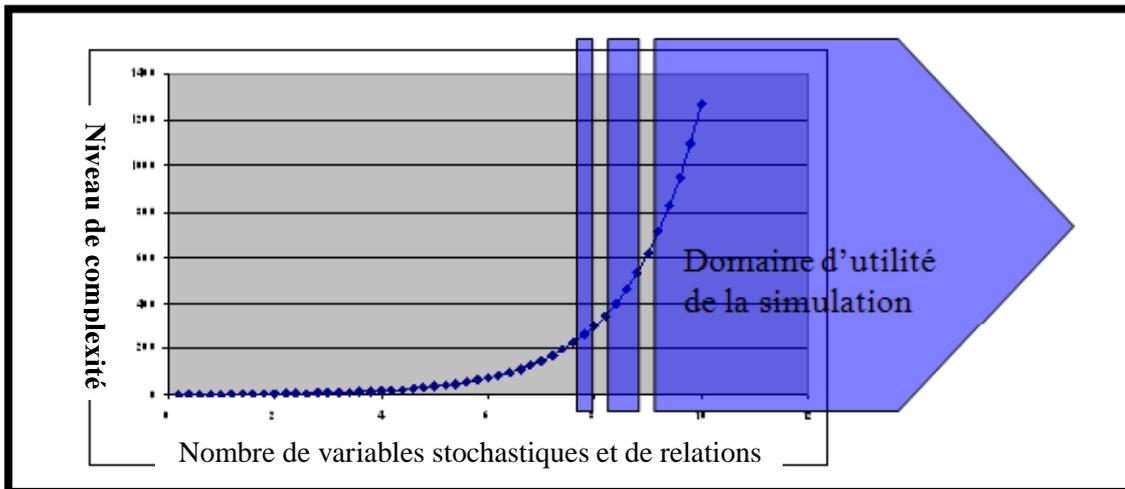


Figure II.6 : Domaine d'application de la simulation.

Parmi les domaines d'application de la simulation :

- Systèmes informatiques ;
- Systèmes industriels ;
- Milieu des affaires (stratégie de marketing, politique des prix, acquisition, transport, planification de la main d'œuvre...) ;
- Gouvernement (militaire, utilisation des terres, soins médicaux...) ;
- Ecologie et environnement ;
- Sociale et comportement ;
- Biosciences (contrôle des maladies, cycle de vie...)

En production, on a recourt à la simulation pour :

- Déterminer le nombre requis de machines et de systèmes de transport pour atteindre un objectif de production ;
- Évaluer les performances de nouvelles politiques de gestion de production ;
- Évaluer les performances de politiques d'ordonnancement ;
- Étudier les politiques d'affectation et de gestion du personnel ;
- Étudier la sensibilité du nombre de kanbans sur les mesures de performance ;
- Analyse financière ou économique d'un projet de construction d'usine ;
- Étudier les algorithmes de pilotage...etc.

V.4. Les étapes des études de simulation

De manière relativement simplifiée, le processus de simulation peut être présenté selon la figure II.7. La modélisation fournit un modèle conceptuel (modèle de connaissance), la programmation fournit un modèle exécutable (modèle d'action), la simulation ou l'expérimentation fournit les résultats obtenus du modèle, l'analyse des résultats permet d'évaluer le processus à modéliser.

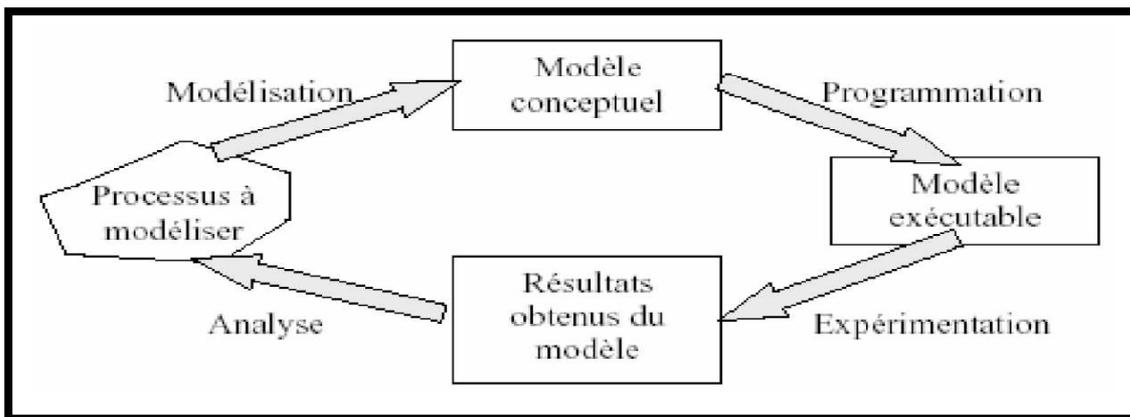


Figure II.7 : Processus simplifié de simulation (Habchi, 2001).

De manière plus détaillée, le processus de simulation peut être éclaté en dix étapes (Pritsker, 1986), (figure II.8).

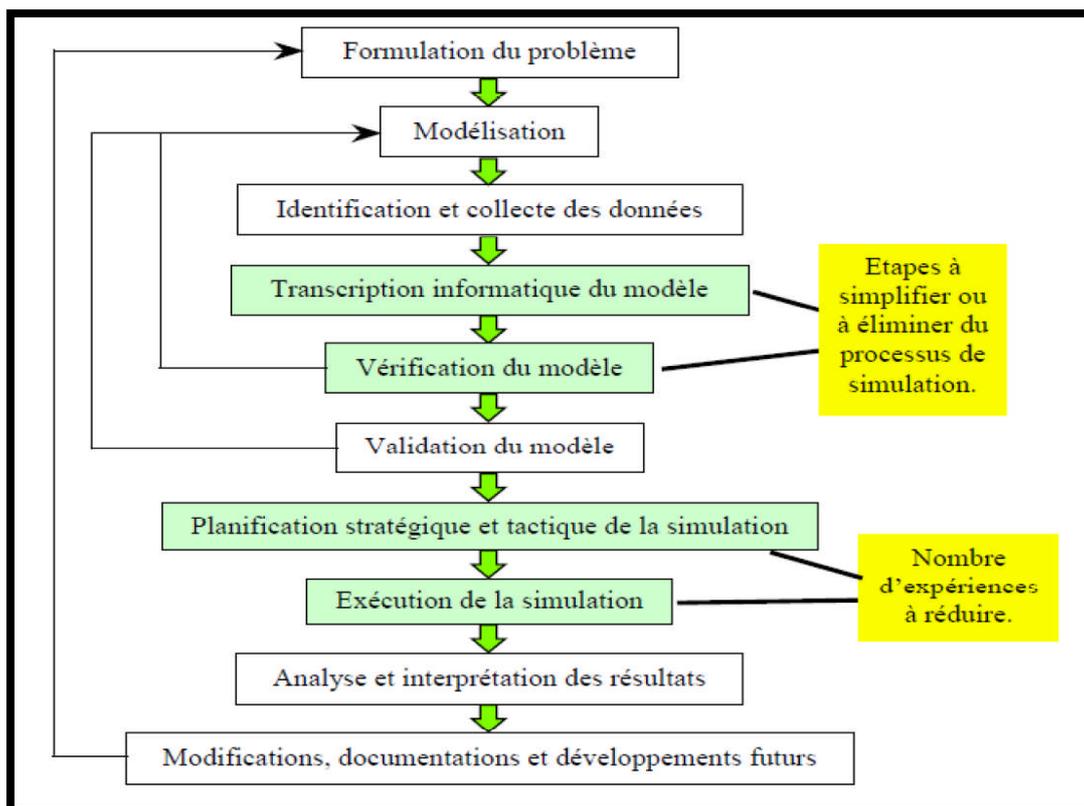


Figure II.8 : Processus de simulation selon (Pritsker, 1986)

V.5. Les avantages et inconvénients de la simulation

a). Les avantages :

La simulation présente de nombreux avantages :

- La procédure de conception du modèle de simulation permet de gagner des connaissances plus précises sur le système analysé et peut même prouver que le système peut opérer de manière différente à ce que tout le monde pense ;
- Une fois le modèle construit, il peut être utilisé à plusieurs reprises ;
- Rôle de *macroscope* et/ou de *microscope* : la simulation permet une vision globale du système étudié, ce qui est particulièrement intéressant quand le système est complexe, et/ou quand la taille réelle est très importante. A l'inverse, la simulation permet de réaliser des zooms importants liés au niveau de détail adapté ;
- La simulation permet d'étudier n'importe quel système sans hypothèses restrictives.
- Outil de *médiation* : l'utilisation d'un simulateur peut servir de négociation, de base de discussion entre intervenants, en illustrant les conséquences des hypothèses en discussion ;
- *Conception, modification et dimensionnement par anticipation* : chacun des choix (gestion, paramétrage, ...) pouvant être réalisé au cours de la conception, suite à une réorganisation, un redimensionnement, une modification, et ayant un impact sur l'exploitation, mérite d'être évalué au préalable grâce à la simulation...

b). Les inconvénients :

On peut citer :

- Les modèles de simulation peuvent être coûteux puisqu'ils nécessitent un temps assez important de construction et de validation ;
- Plusieurs exécutions sont souvent requises, ce qui implique des coûts de calcul assez importants (ce n'est plus le cas avec l'évolution des technologies) ;
- La simulation nécessite des analystes assez familiers avec cette méthodologie ;
- Les résultats peuvent être mal interprétés, notamment lorsque la phase transitoire n'est pas prise en compte ;
- Des facteurs humains et technologiques peuvent être ignorés, mauvais choix du niveau d'abstraction ;
- Pas de bons résultats si les données sont erronées ou si le modèle n'est pas correct.

Il faut tout de même souligner pour ne pas induire en erreur que les avantages de la simulation dépassent ses inconvénients.

VI. Présentation du logiciel ARENA 12

ARENA est un logiciel dédié de modélisation et de simulation des systèmes de production, des procédés, des chaînes logistiques, de la distribution, des stocks...par l'**approche processus**¹. Il présente une facilité d'utilisation en étant un simulateur de haut niveau, et une flexibilité de la modélisation en utilisant le langage de simulation SIMAN et le script Microsoft Visual Basic (VBA).

Les composants d'ARENA sont programmés en SIMAN, ce qui permet de créer ses propres blocs, ce qui améliore la flexibilité du simulateur. De plus, le simulateur comprend des animations graphiques, des courbes et de différents composants d'analyse des résultats.

Dans n'importe quel projet de simulation, que se soit une analyse d'une chaîne logistique ou bien une nouvelle politique de gestion d'atelier manufacturier ou encore une nouvelle disposition des serveurs ou guichets dans une mairie, on passe nécessairement par les cinq étapes suivantes avec le logiciel ARENA (de manière non séquentielle) :

- Créer le modèle de base ;
- Introduire toutes les informations nécessaires au modèle créé ;
- Simuler le modèle ;
- Analyser les résultats de la simulation ;
- Sélectionner la meilleure alternative.

VI.1. Exemple d'un modèle (de simulation) ARENA

Dans cette partie de travail, on va ouvrir un modèle de simulation déjà existant, l'utiliser pour explorer ARENA de manière à ce qu'on se familiarise avec, et du coup on en profite pour introduire la terminologie de base d'Arena.

a). L'environnement de travail :

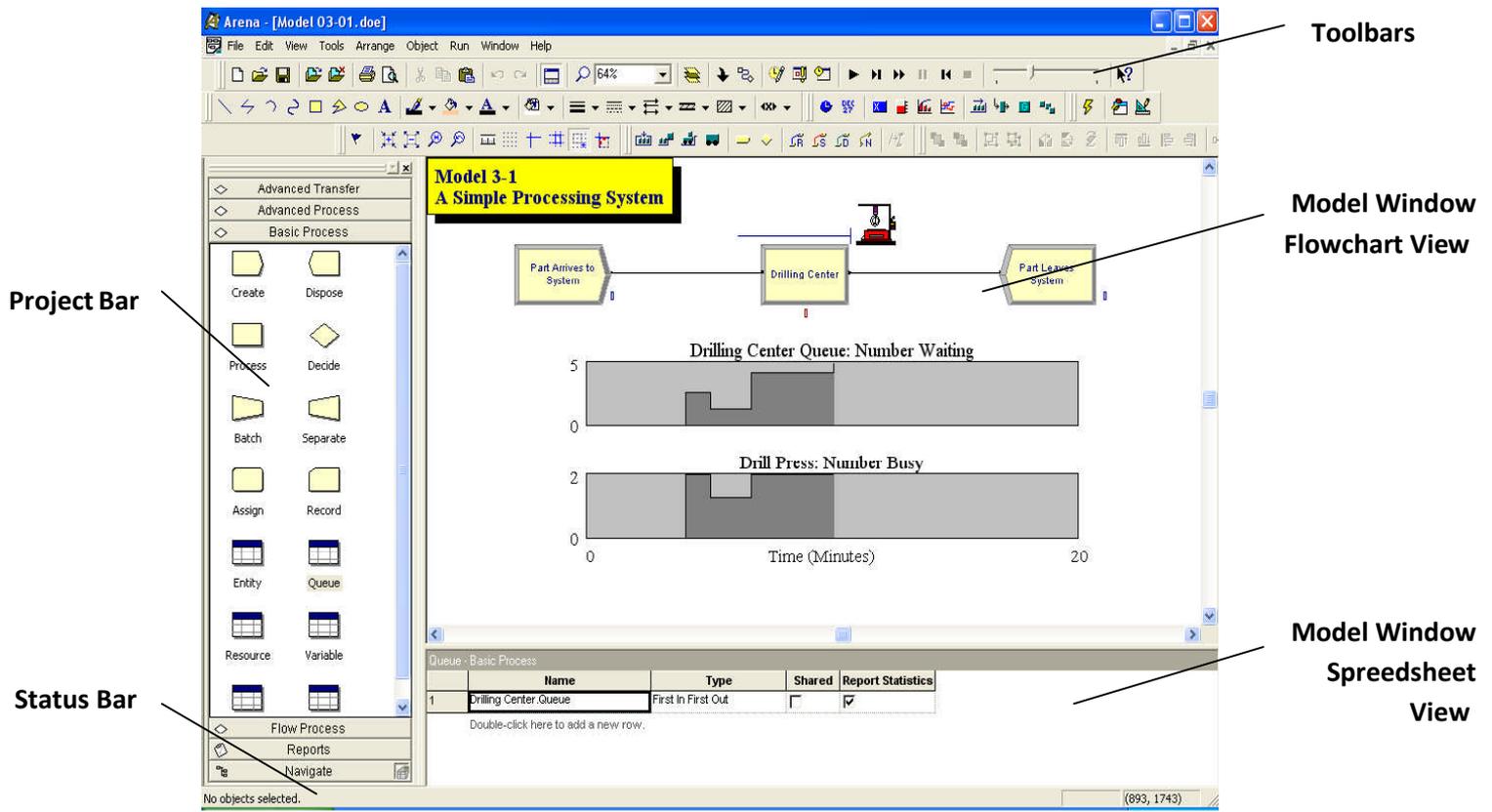


Figure II.9 : Interface graphique d’Arena 12 pour model 3-1.

On peut ouvrir le modèle existant nommé *model 03-01.doe* (.doe est l’extension des modèles sous ARENA) via *File > Open* ou juste en cliquant sur l’icône , une fenêtre apparait, si le fichier n’apparait pas vous chercher le dossier *Book Examples* du dossier *Arena*, vous double cliquez sur le fichier en question. Une interface ressemblant à la figure II.9 devrait alors s’afficher. A présent, nous appelons ce modèle Model 3-1.

b). La structure de base de la fenêtre d’Arena :

Comme il est constaté, l’interface d’Arena pour le modèle ouvert se compose de plusieurs parties distinctes mais complémentaires :

i). La fenêtre du modèle (*Model Window*) :

C’est là que le modèle est bâti, elle est divisée en deux importantes parties : partie organigramme (*Flowchart View*) et partie tableur (*Spreadsheet View*).

— Organigramme :

C’est à ce niveau que le modèle est « dessiné ». Avec les différents blocs, les connections, les animations et les graphiques. Il contient le fichier dit "fichier modèle" avec l’extension

".mod" qu'on peut visualiser via *Run > SIMAN > View*. Ce fichier contient toutes les informations qui concernent la partie statique du modèle. Dans notre cas c'est les blocs *Create-Process-Dispose*.

— **Tableur :**

Il contient les informations concernant les différents attributs et variables, des files d'attente et des ressources, des plannings et des ensembles de blocs. Il contient le fichier dit "fichier expérimental" avec l'extension ".exp" qu'on peut visualiser via *Run > SIMAN > View*. Ce fichier contient les informations de la partie dynamique du modèle. Dans notre cas c'est les tableaux *Entity-Queue-Resource*.

ii). **La bibliothèque d'objets :**

Dans la partie gauche de la fenêtre on trouve la bibliothèque d'objets *Project Bar* qui contient les objets programmés "*Templates*" principalement "*Basic Process*" qui contient les blocks fondamentaux (*modules*) utiles pour construire une large variété de modèles de simulation (fichier modèle et fichier expérimental). On y trouve les modules :

Create, Dispose, Process, Decide, Batch, Separate, Assign, Record, Entity, Queue, Resource, Variable, Schedule and *Set* (qui seront détaillés un peu plus tard).

iii). **Une barre d'outils (Toolbar) :**

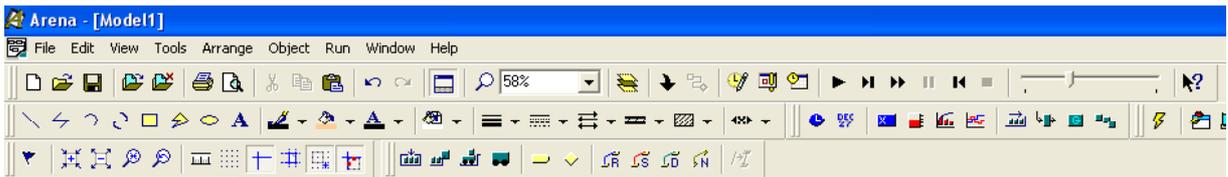


Figure II.10 : La barre d'outils.

Comportant les fonctionnalités (sous barres) :

- *Standards* : nouveau, ouvrir, enregistrer, copier, coller, revenir, répéter, imprimer, zoom, attacher de nouveaux *templates*, détacher des *templates*, les connections entre blocs ainsi que les boutons de commande de la simulation : début, pause, arrêt, avance rapide, retour, etc.
- *Draw* (de dessin)
- *Animate* (d'animation)...

iv). **Une barre d'état (Status Bar) :**

Elle donne en gros le temps écoulé de la simulation, la date de la simulation, le numéro de l'instance et les coordonnées du pointeur de la souris sur l'espace de dessin.

VI.2. L'application VBA de Microsoft sous ARENA

Les concepteurs du logiciel ARENA ont intégré les fonctionnalités du VBA dans le but d'asservissement du modèle de simulation et pour plus de flexibilité.

Le recours à cette automation s'effectue de deux manières :

- En faisant glisser le bloc *VBA* dans le *template Blocks* ;
- Ou bien en recourant à la barre d'outils (ou *Alt+F11*).

La connaissance de base qu'il faut avoir sur le VBA sous ARENA est comment VBA intervient-il dans le modèle de simulation et quand ?

VBA entre en interaction avec le modèle ARENA (variables, attributs...) en déclarant un objet dit SIMAN (*Alt+F11*) de la manière suivante :

```
Dim oSIMAN As Arena.SIMAN  
Set oSIMAN = ThisDocument.Model.SIMAN
```

Le *ThisDocument* est le fichier où les scripts sont tapés.

Le temps (les événements) d'intervention des scripts sont en gros de trois types :

- Les évènements d'avant exécution de la simulation : DocumentOpen, DocumentSave...
- Les évènements d'initialisation de la simulation : RunBegin, RunBeginSimulation...
- Les évènements lors de la simulation : VBA_Block_Fire, OnKeyStroke...

VI.3. Illustration (model 3- 1)

i). Pour créer le fichier modèle, il suffit de glisser les modules de *Project Bar* vers *Flowchart View Window*, ainsi pour créer notre modèle, on glisse les trois modules *Create*, *Process* and *Dispose* puis on les connecte via *Object > Connect*, comme montré dans la figure ci-dessous.



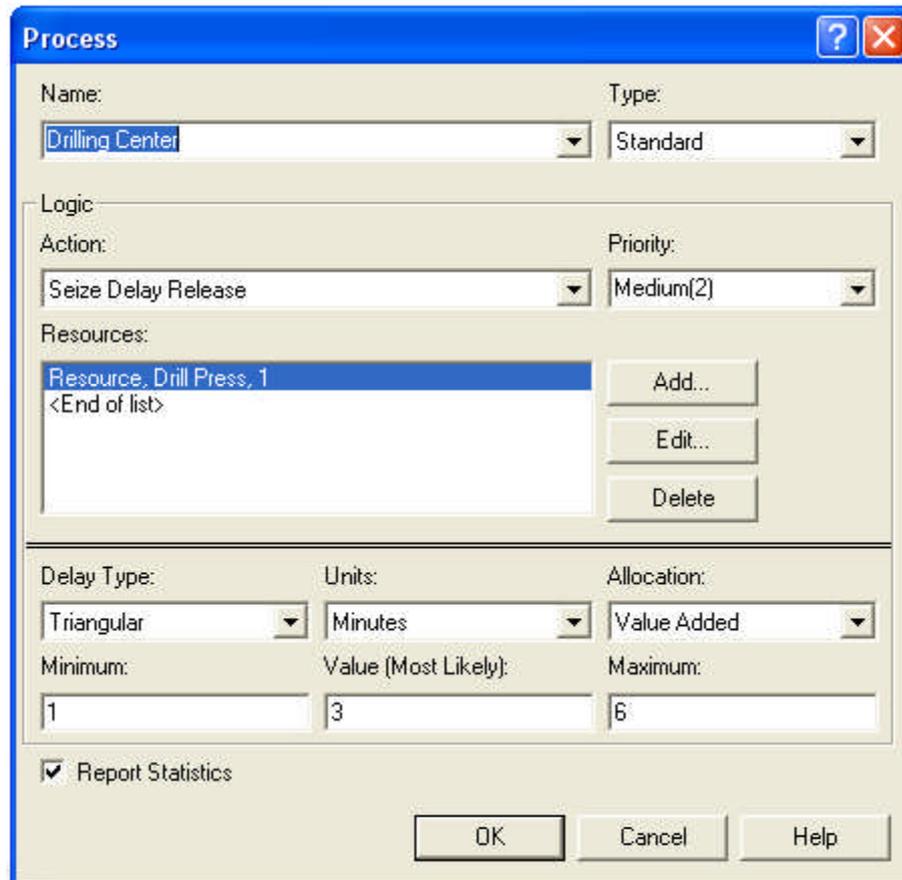
Figures II.11 : Création du fichier

On renseigne après les informations du modèle dans chaque module pour le définir, on détaillera pour le module *Create* puis c'est la même chose pour les autres.

Name:		Entity Type:	
Part Arrives to System		Part	
Time Between Arrivals			
Type:	Value:	Units:	
Random (Expo)	5	Minutes	
Entities per Arrival:	Max Arrivals:	First Creation:	
1	Infinite	0	

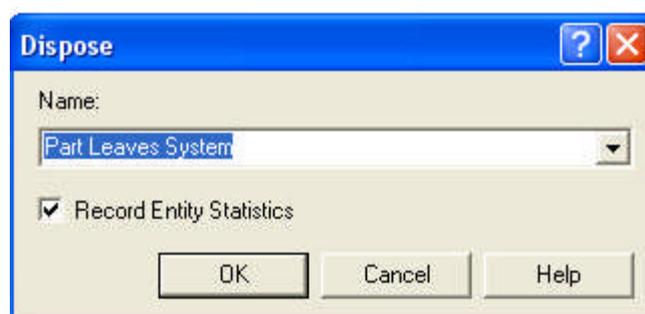
Figures II.12: Module *Create*.

Comme il est constaté, le bloc est nommé *Part Arrives to System*, le type d'entité est *Part*, la création d'entités suit une *loi exponentielle* de moyenne *5 minutes*, une entité est créée à la fois dont la première est créée à l'instant *0*, le nombre d'arrivées est *infini*.



Figures II.13: Module *Process*.

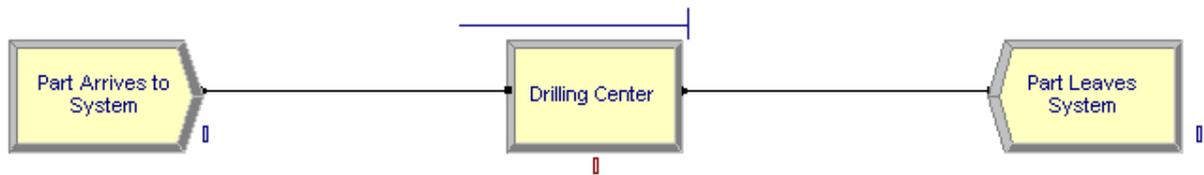
Pour le module *Process*, nommé *Drilling Center* (Centre de perçage), il permet la saisie de la machine (Ressource Drill Press de capacité 1) pour traitement de la pièce puis la libération de cette dernière après traitement Seize-Delay-Release, la durée de traitement suit une loi triangulaire de valeur minimale 1 mn, moyenne 3 mn (la plus utilisée) et maximale 6 mn. Pour plus d'information il suffit de cliquer sur le bouton *Help*.



Figures II.14: Module *Dispose*.

Le module *Dispose* permet de détruire les entités ayant subi tous les processus du modèle, avant la destruction, des statistiques définies par défaut sont enregistrées pour n'importe quelle entité à détruire.

Le fichier modèle est à présent :



Figures II.15: le modèle.

ii). Pour créer le fichier expérimental, il suffit maintenant d'éditer (on ne peut pas les faire glisser dans la fenêtre) les modules sous forme de tableaux. On commence par le module *Entity* figure ci-dessous :

Entity - Basic Process							
	Entity Type	Initial Picture	Holding Cost / Hour	Initial VA Cost	Initial NVA Cost	Initial Waiting Cost	Initial Tran
1	Part	Picture.Blue Ball	0.0	0.0	0.0	0.0	0.0

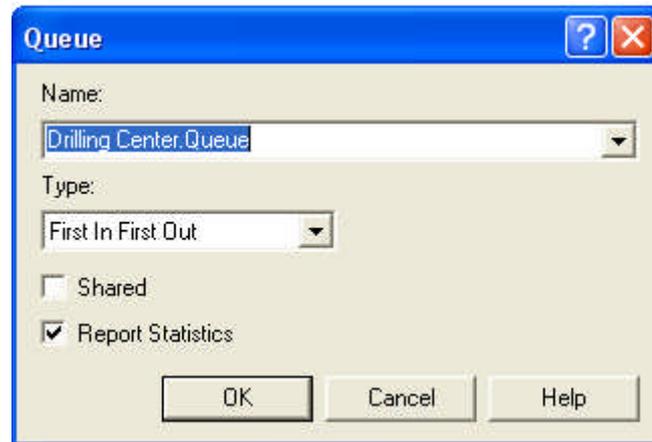
Double-click here to add a new row.

Figures II.16.a : Module *Entity*.

On peut remplir ce tableau d'une manière similaire au cas précédent en cliquant sur le bouton droit et en choisissant *Edit via Dialog* comme le montre la figure ci-dessous :

Figures II.16.b : Module *Entity*.

Ce module permet de créer et définir l'image représentative de l'entité utile dans la partie animation du modèle, c'est l'image *Picture. Blue Ball* qui est sous la forme d'un disque bleu.

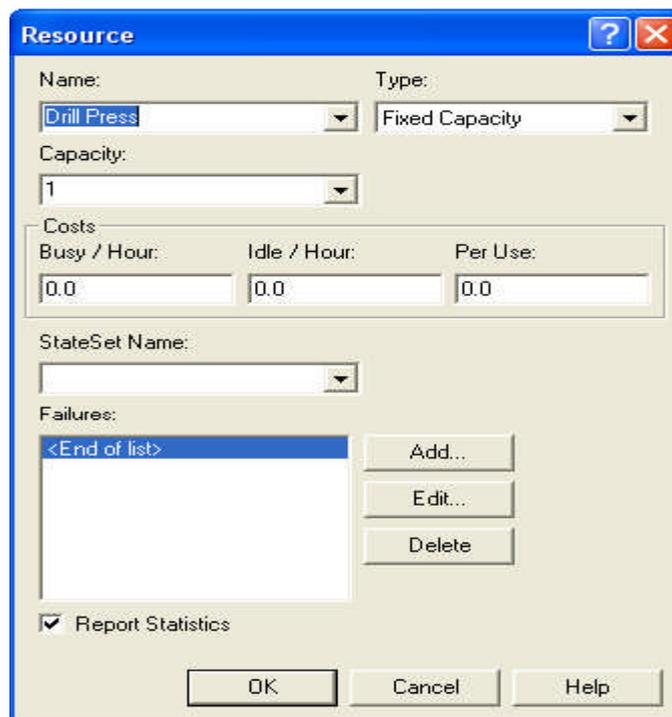


Figures II.17: Module *Queue*.

Le module *Queue* ci-dessus nous permet de rattacher une file d'attente à la perceuse en cas où le temps d'arrivée pièces est moins important que le temps de traitement.

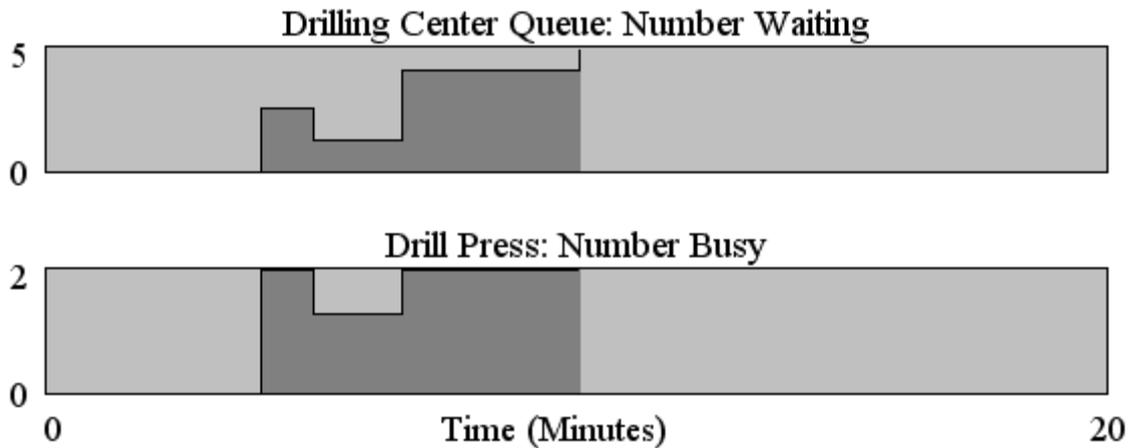
Et sans oublier la déclaration des propriétés de la ressource (Perceuse) qui est possible grâce au module *Resource*.

iii). Une fois le modèle élaboré, on peut y intégrer des animations graphiques, dérouler au cours de la simulation les courbes des variables définies par défaut comme $NQ(\text{Drilling Center.Queue})$ qui est le nombre de pièces dans la file d'attente à l'instant t de la simulation, ou encore suivre l'état d'occupation de la machine $NR(\text{Drill Press})$. Voir la figure II.18.



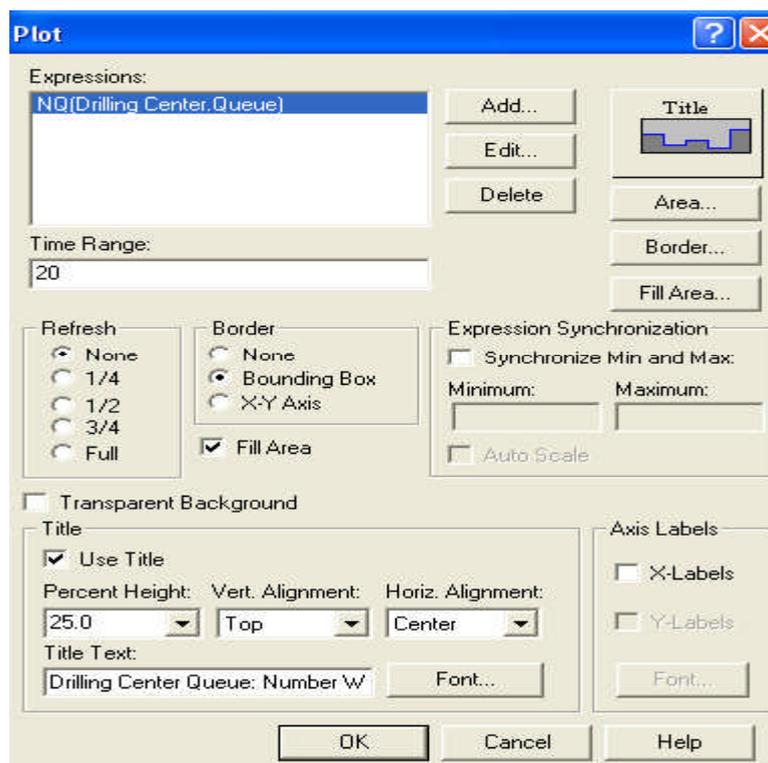
Figures II.18: Module *Resource*.

Pour définir l'image de la ressource, on sélectionne l'icône *Resource* dans la barre d'outils *Animate*, on peut ainsi définir ces états, choisir l'image qui nous convient...



Figures II.19.a : Animation des variables.

On peut ajouter des courbes via la barre d'outils *Animate*.

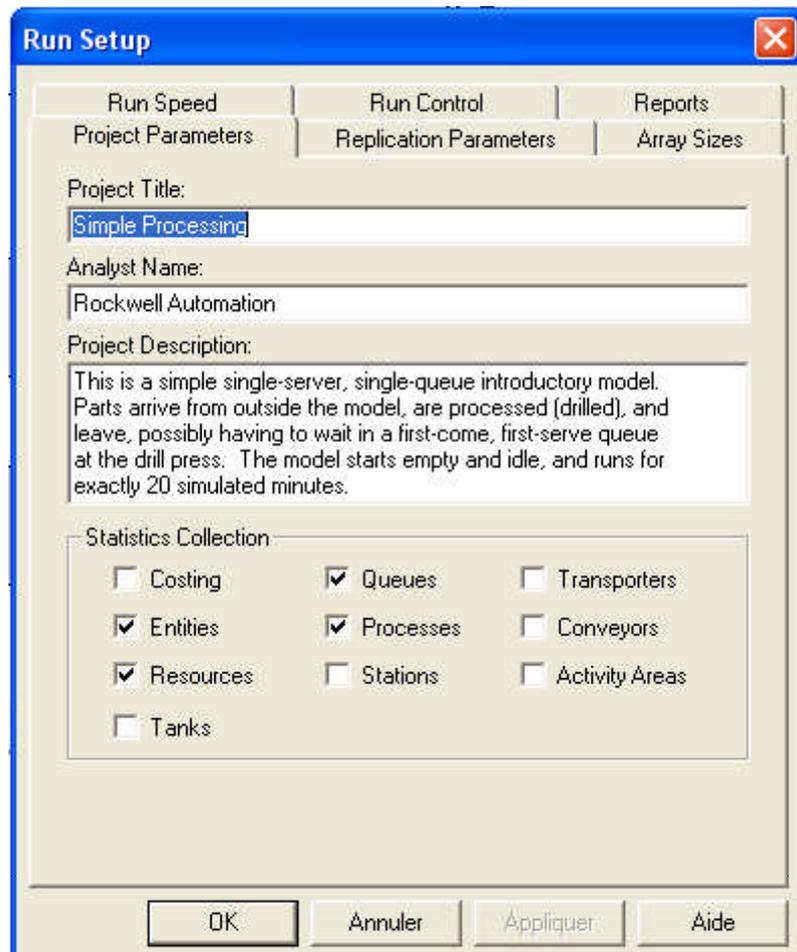


Figures II.19.b : Animation des variables.

La figure ci-dessus est générée à partir de l'icône *Plot* de barre d'outils *Animate*. Pour définir au mieux le modèle, des possibilités de dessin et d'écriture sont offertes via la barre d'outils *Draw*.



Pour finir le modèle, on passe à la partie où sont définis les paramètres d'exécution de ce dernier, on peut le faire via *Run > Setup*.



Figures II.20 : La fenêtre *Setup*.

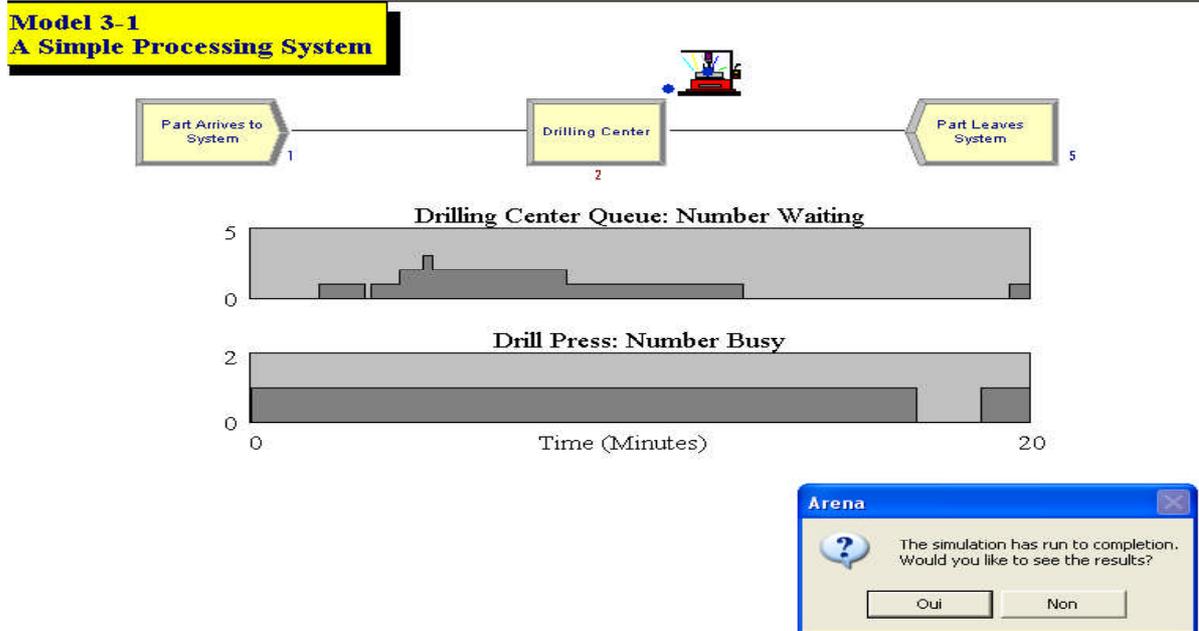
Cette fenêtre nous permet de renseigner le titre du modèle, l'analyste (celui qui a élaboré le modèle), des précisions sur le modèle, les statistiques à enregistrer, renseigner les paramètres des simulations (nombre de simulations à effectuer, définir l'unité du temps...)...etc.

iv). Exécuter le modèle :

Pour exécuter le modèle il suffit d'aller dans la barre d'outils *Standard* et cliquer sur le bouton *Go*.

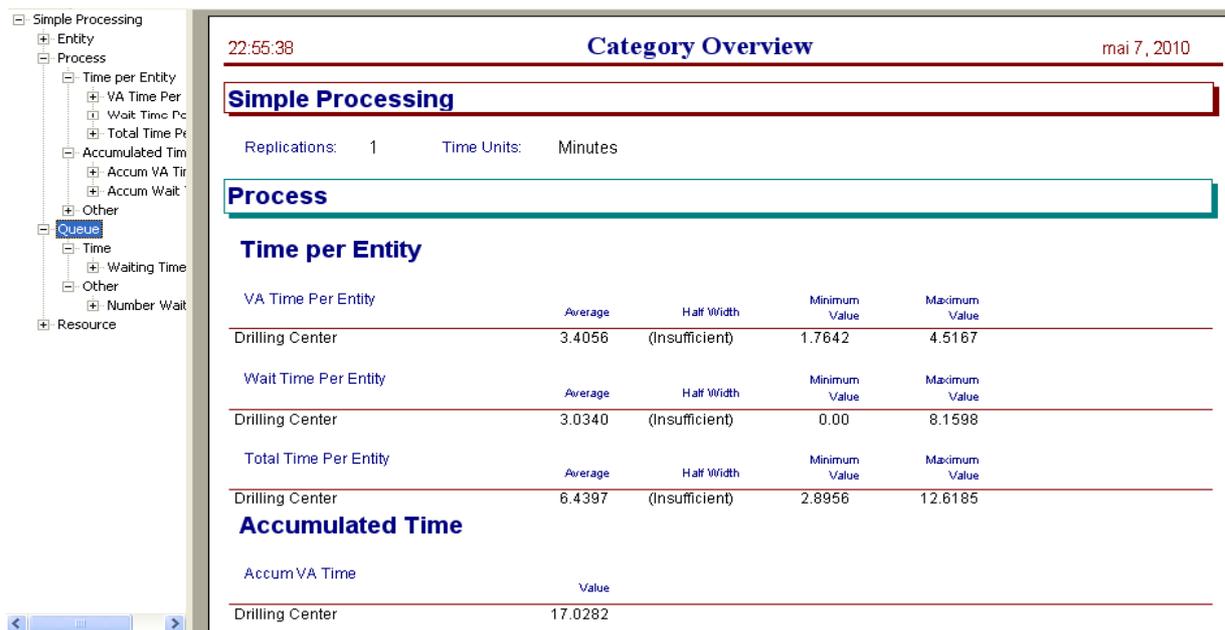


Une fois le modèle exécuté, une boîte de dialogue s'affiche nous informant que la simulation est terminée et nous demandant si nous voulons ou non voir le rapport de la simulation, qui n'est d'autre qu'une collection de statistiques comme le nombre d'entités créées, traitées, les défaillances de la machines s'il y'en a eu...



Figures II.21 : Le modèle après exécution.

La fenêtre du rapport a la forme suivante :



Figures II.22 : Le fichier Rapport.

Pour plus de détails sur la simulation avec Arena, consulter l'aide du logiciel et/ou les modèles préconstruits par ce dernier et/ou le livre édité pour ce but (Kelton, 2004).

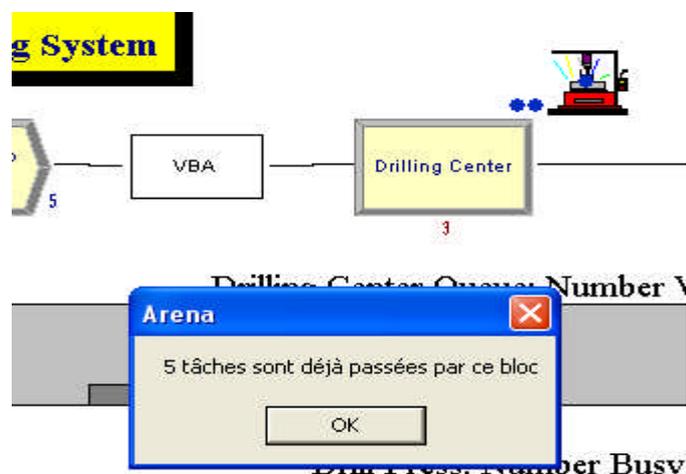
v). Partie VBA :

Dans ce qui suit, Nous allons perfectionner le modèle *Model 3-1* en lui rajoutant un bloc VBA. La figure qui suit montre comment est fait le script, on expliquera ensuite à quoi il sert et c'est à titre d'exemple.

```
'Déclarer le modèle oSIMAN comme un objet modèle
Dim oSIMAN As Arena.SIMAN
Private Sub ModelLogic_RunBegin()
'Affichage d'un message « Initialisation du modèle »
MsgBox " Initialisation du modèle "
End Sub
Private Sub ModelLogic_RunBeginSimulation()
'Ajouter les variables d'état au modèle oSIMAN à chaque début de simulation
Set oSIMAN = ThisDocument.Model.SIMAN
End Sub
Private Sub ModelLogic_RunBeginReplication()
'À chaque début de replication, un message « début de la réplication » est affiché
MsgBox " début de la réplication "
End Sub
Private Sub VBA_Block_1_Fire()
'Déclarer c comme un entier statique pendant toute la simulation
Static c As Integer
c = c + 1
If c = 5 Then
MsgBox "5 tâches sont déjà passées par ce bloc"
End If
'Afficher un message donnant le temps de passage d'une entité à travers le bloc VBA
MsgBox "TNOW = " & oSIMAN.RunCurrentTime
End Sub
```

Figures II.23 : Exemple de Script VBA sous ARENA.

Les écritures en vert sont les explications de chaque partie du programme VBA. En exécutant ce programme, l'une des fenêtres qui s'affichera est la suivante :



Figures II.24 : Exemple d'exécution du script VBA.

- 1. RunBegin**
2. ARENA teste et initialise le modèle
- 3. RunBeginSimulation**
- 4. RunBeginReplication**
5. ARENA exécute la réplication
- OnKeyStroke**
- UserFunction**
- 6. RunEndReplication**
- 7. RunEndSimulation**
8. ARENA termine la simulation
- 9. RunEnd**

Figure II.25 : Les moments d'intervention de VBA.

La fonction *ModelLogic_VBA_Block_Fire* s'exécute à chaque fois qu'une entité passe par le bloc VBA concerné. (Voir l'exemple ci-dessus).

VII. Conclusion

Dans le cadre de ce chapitre, nous avons présenté et décrit la cellule flexible d'assemblage et justifié l'approche de résolution retenue.

Du fait du caractère NP-difficile au sens fort des problèmes d'ordonnancement considérés, nous avons opté pour une approche de résolution empirique dont la simulation est le support principal. Pour ce faire, on a présenté les concepts de base de cette dernière.

Dans le chapitre suivant, nous allons concevoir un modèle de simulation de la cellule étudiée et par la suite nous allons le valider.

Chapitre III

MODELE DE SIMULATION

*“ Le commencement de toutes les sciences,
c'est l'étonnement de ce que les choses
sont ce qu'elles sont ”*

[Aristote]

Chapitre III : Modèle de simulation

I. Introduction

II. Rappel du problème

III. Mise en œuvre du processus de simulation

III.1. Modélisation de la cellule d'assemblage

III.1.1. Modélisation par l'approche processus

III.2. Construction du modèle ARENA et transcription informatique du modèle

III.2.1. Modélisation statique de l'atelier d'assemblage

III.2.2. Interface d'expérience

III.2.3. Interface des résultats

III.2.4. Modélisation dynamique de l'atelier d'assemblage

III.3. Vérification et validation du modèle

IV. Conclusion.

I. Introduction

Après avoir exposé notre démarche dans le chapitre précédent et justifié l'utilisation de l'approche empirique, précisément la simulation comme outil de résolution de notre problématique, nous entamons dans ce présent chapitre l'essentiel de notre travail ; l'élaboration du modèle de simulation. Pour ce faire, on a adopté la démarche décrite par Pritsker (Pritsker, 1986) à savoir les six premières étapes du processus, où on commencera par un rappel du problème, puis la modélisation et la transcription informatique du modèle, enfin la vérification et la validation de ce dernier.

II. Rappel du problème

Dans le cadre de ce mémoire, on étudie le comportement d'une cellule flexible d'assemblage dont la typologie est circulaire, et dont les problèmes d'ordonnancement considérés sont du type Job-shop et Open-shop.

Les pièces nécessaires à l'assemblage d'un produit fini sont contenues sur des palettes. En cas de conflit lors de la sortie d'une pièce d'une station donnée, la priorité est donnée à celles circulant sur le convoyeur.

Sur la base du rappel du problème précédant, on procède à la caractérisation de la cellule selon les éléments suivant :

- Le convoyeur en boucle où les palettes y circulent à la même vitesse et sans arrêt ;
- Les stations annexées au convoyeur tout en assurant l'entrée des palettes (solidaires aux pièces à assembler) ;
- Les pièces et les palettes présentes sur le convoyeur ;
- Les règles de priorité choisies au niveau des stations.
- Les files d'attente d'entrée et de sortie à capacités limitées dans les stations.
- Les conditions d'entrée dans les stations suivant une logique Open-Shop ou Job-Shop.
- Les conditions de sortie de stations afin de donner la priorité aux palettes circulantes sur le convoyeur.

Remarque :

Un autre type d'ordonnancement a été introduit, il s'agit d'un modèle collaboratif dont le protocole de coopération est inspiré de (Baba Aissa et Maouassa, 2008), il est implémenté

seulement pour 4 stations afin de vérifier la stabilité des résultats fournis dans (Baba Aissa et Maouassa, 2008).

❖ Description du protocole de coopération :

L'arrivée d'une pièce au niveau d'une *station i* déclenche le protocole de coopération par l'envoi d'une demande d'information (**Query**) à l'ensemble des autres stations non encore visitées par la pièce. Cette requête porte sur l'état des files d'attente des stations interrogées. A partir des renseignements obtenus, s'il s'avère que la station initiatrice n'est pas la plus **adaptée** pour le traitement de la pièce selon un critère préalablement établi, celle-ci envoie des demandes d'engagements (**Request**) à l'ensemble des stations non visitées. La station initiatrice reçoit alors au moins un message d'acceptation (**Agree**).

Néanmoins, si plusieurs stations acceptent de s'engager, la station initiatrice déclenche alors une procédure de recherche de la station la plus adaptée pour l'exécution de la tâche c'est-à-dire la station ayant la file d'attente la moins saturée et la plus proche sur le convoyeur de l'emplacement actuel de la pièce. La station sélectionnée reçoit alors un message de confirmation (**Confirm**) de la part de la station initiatrice et réactualise son échéancier d'exécution. En cas de désengagement de la station sélectionnée, un message d'annulation (**Cancel**) est envoyé par celle-ci à l'ensemble des participants.

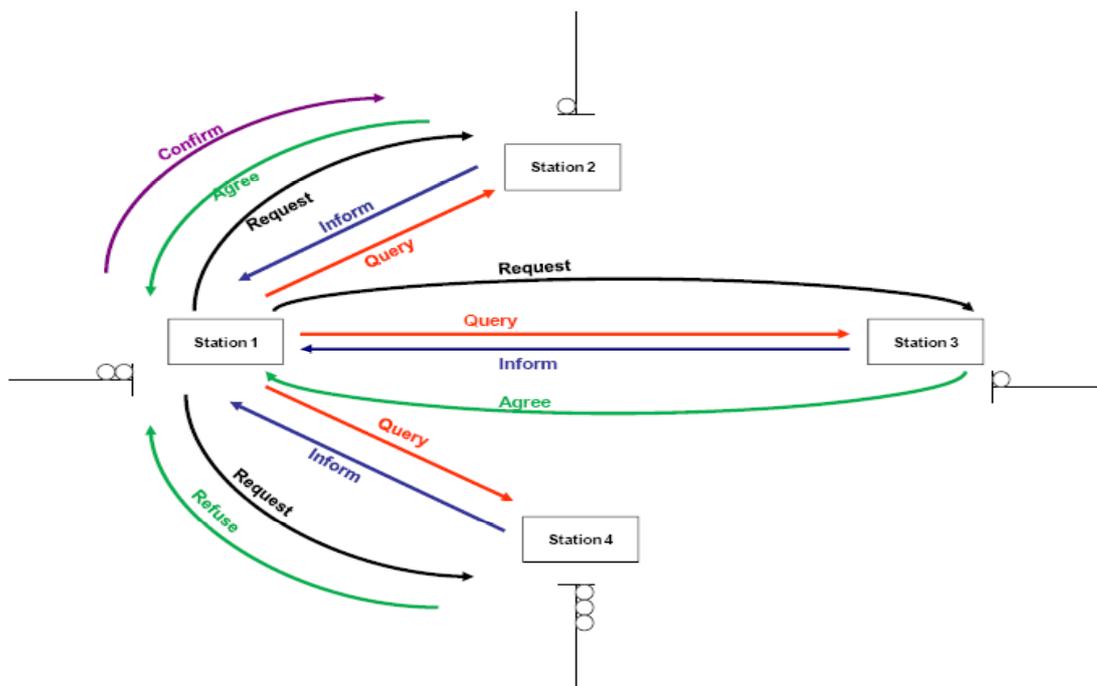


Figure III.1 : Philosophie générale du protocole de coopération

❖ Les étapes du protocole de coopération :

Dans ce qui suit, on décrira avec plus de détails les différentes étapes du processus de coopération mis en œuvre dans le cadre du système d'ordonnancement de la cellule d'assemblage. Ce protocole permettra à la fois de spécifier la répartition des capacités décisionnelles entre les différentes ressources et de fixer les règles d'interaction afin de réaliser une coopération entre elles.

Etape 1 : Envoi de demandes d'information

A l'arrivée d'une pièce à l'entrée d'une *station i*, celle-ci envoie une demande d'information (**Query**) à l'ensemble des stations non encore visitées par la pièce. Cette demande d'information porte sur l'état des files d'attente de ces stations. Lorsqu'une *station j* reçoit une demande d'information, elle exécute aussitôt une **procédure de calcul** pour évaluer le nombre de pièces dans sa file d'attente d'entrée $NFAE_j$ à l'instant $TNOW + T_{ij}$, tel que T_{ij} représente le temps de transfert de la *station i* vers la *station j*. Ce calcul est ainsi effectué afin de déterminer l'état de la FAE de la *station j* au moment de l'arrivée de la pièce dans l'éventualité qu'elle serait plus adaptée que la *station i* pour exécuter la tâche qui lui revient.

Etape 2 : Réception des messages d'information

Dès que les données sont disponibles au niveau de la *station i*, elle teste alors la condition suivante :

$$NFAE_i(TNOW) = \text{Min} \{ \{ NFAE_j(TNOW + T_{ij}) \}_j, NFAE_i(TNOW) \}$$

tel que I est l'ensemble des stations non visitées.

- Si cette condition est vérifiée alors la *station i* se charge de traiter la pièce et l'envoie à la fin du traitement vers la station la plus proche non encore visitée.
- Sinon : Elle envoie une requête de type demande d'engagement aux autres stations participantes (**Request**).

En résumé, l'objectif de cette deuxième étape est de déterminer si la *station i* est la plus adaptée pour traiter la pièce c'est-à-dire la station ayant la file d'attente la moins saturée et donc celle qui engendrera un temps d'attente minimal pour la pièce.

Etape 3 : Collaboration des stations pour minimiser le temps d'attente et de transfert

Les *stations j*, capables de prendre en charge la pièce, envoient un message d'acceptation à la *station i* (**Agree**).

- Si la *station i* reçoit plusieurs messages d'acceptation, elle fait appel à une procédure de recherche de la station la plus adaptée c'est-à-dire la moins saturée et la plus proche sur le convoyeur de l'emplacement actuel de la pièce (la pièce est toujours à l'entrée de la *station i*). La pièce est alors transférée vers la station sélectionnée.
- Sinon la pièce est envoyée vers l'unique *station j* ayant accepté la requête.

En résumé, l'objectif de cette étape est d'assurer la meilleure affectation possible des pièces du point de vue du temps d'attente ; en choisissant la station dont la file d'attente d'entrée est le moins saturée, et du temps de transfert ; en sélectionnant la station la plus proche sur le convoyeur qui satisfait au critère précédent.

III. Mise en œuvre du processus de simulation

Cette section décrit la mise en œuvre du processus de simulation pour la cellule flexible d'assemblage étudiée.

III.1. Modélisation de la cellule d'assemblage

De nombreux logiciels, et c'est le cas de SIMAN/ARENA, privilégient l'approche processus en tant qu'approche de modélisation. Un processus est une séquence de transformation que subit une entité pendant son séjour dans le système de production. Une entité est un objet dont l'état est modifié (le plus souvent) par le processus qu'elle subit. Les changements d'états nécessitent souvent l'utilisation de ressources. Un modèle est constitué par un ensemble de processus qui interagissent entre eux.

III.1.1. Modélisation par l'approche processus

La construction du modèle logique de la cellule d'assemblage à l'aide du logiciel Arena peut dès lors s'articuler en plusieurs étapes successives :

a) Identification des entités

L'entité considérée dans ce système est l'association des entités (représentant les pièces à assembler) solidaires à leurs palettes.

b) Identification des processus

On a recensé quatre catégories de processus :

- Un processus d'entrée ;
- Un processus de transfert entre stations ;
- Un premier processus de transformation : à chaque station considérée selon la configuration de l'expérience (4 stations, 5, ..., jusqu'à 10 stations) ;
- Un processus de sortie.

c) Identification des ressources

- **Les ressources de transformation** : quatre à dix robots, chacun dans une station portant le même indice.
- **Les ressources de stockage** : des files d'attente, une à l'entrée du système (Qentrée), deux files d'attentes propres à chaque station : une d'entrée (robotpr.Queue, robotde.Queue, robottr.Queue, robotq.Queue, robotcq.Queue, robotsx.Queue, robotst.Queue, robotht.Queue, robotnf.Queue et robotdx.Queue) et une autre de sortie (QS1, QS2, QS3, QS4, QS5, QS6, QS7, QS8, QS9 et QS10).
- **Les ressources de transport** : un convoyeur en boucle reliant toutes les stations, dix petits convoyeurs reliant l'entrée de chaque station à son robot et enfin dix autres petits convoyeurs reliant le robot de chaque station à sa sortie.

d) Identification des attributs et des variables :

Ils caractérisent le système dans son évolution au cours du temps :

Les Attributs ce sont des variables spécifiques à une entité donnée qu'elle transporte avec elle pendant sa vie dans le système (type d'entité, gamme opératoire, date d'entrée dans le système, etc.). Ces variables sont locales et interviennent dans le pilotage du système (exemple : la gamme de fabrication d'une pièce va déterminer la décision d'envoi de cette pièce sur les machines de l'atelier).

Les variables de simulation : celles-ci sont au contraire connues de tous les processus (ce sont des variables globales). Elles concernent en général les durées de séjour des entités dans le système, les taux d'utilisation des ressources et le niveau des encours.

Pour le système étudié, on a défini les attributs et les variables suivants :

- do1 : durée opératoire de l'entité par rapport au premier robot.
- do2 : durée opératoire de l'entité par rapport au deuxième robot.
- do3 : durée opératoire de l'entité par rapport au troisième robot.
- do4 : durée opératoire de l'entité par rapport au quatrième robot.
- do5 : durée opératoire de l'entité par rapport au cinquième robot.
- do6 : durée opératoire de l'entité par rapport au sixième robot.
- do7 : durée opératoire de l'entité par rapport au septième robot.
- do8 : durée opératoire de l'entité par rapport au huitième robot.
- do9 : durée opératoire de l'entité par rapport au neuvième robot.
- do10 : durée opératoire de l'entité par rapport au dixième robot.
- dtot : cet attribut représente la somme de toutes les durées opératoires à effectuer par l'entité à travers la cellule. Il est mis à jours à chaque sortie d'un robot (soustraction de la durée effectuée).
- picture : associe une image propre à chaque type de pièce.
- creation : représente le temps de création de l'entité.

Et pour les variables de simulation (Arena):

- choose : prend les valeurs (1, 2, 3 et 4) selon l'ordonnancement choisi (Open-shop, Job-shop, Flow shop et l'approche distribuée) respectivement.
- FAE : prend la valeur de capacité des files d'attente d'entrée de chaque station ^[1].
- nbrm : représente le nombre de machine (stations) de 4 jusqu'à 10.
- NBR : représente le nombre de instances de chaque expérience
- CMAXX : représente le Makespan (temps de séjour total d'une commande).

[1] : Dans le cas Flow Shop, FAE= 1000 afin que la condition (file d'attente non saturée) soit toujours vérifiée ce qui nous garantit un comportement Flow Shop.

III.2. Construction du modèle ARENA et transcription informatique du modèle

On a conçu un programme sous Rockwell ARENA 12. Dans ce qui suit, on décrira les différentes parties du modèle Arena pour la cellule d'assemblage étudiée. Ce modèle comporte deux parties : une partie **statique** qui décrit les différents processus identifiés et une partie **dynamique** qui permet d'animer les différentes composantes du système.

III.2.1. Modélisation statique de l'atelier d'assemblage

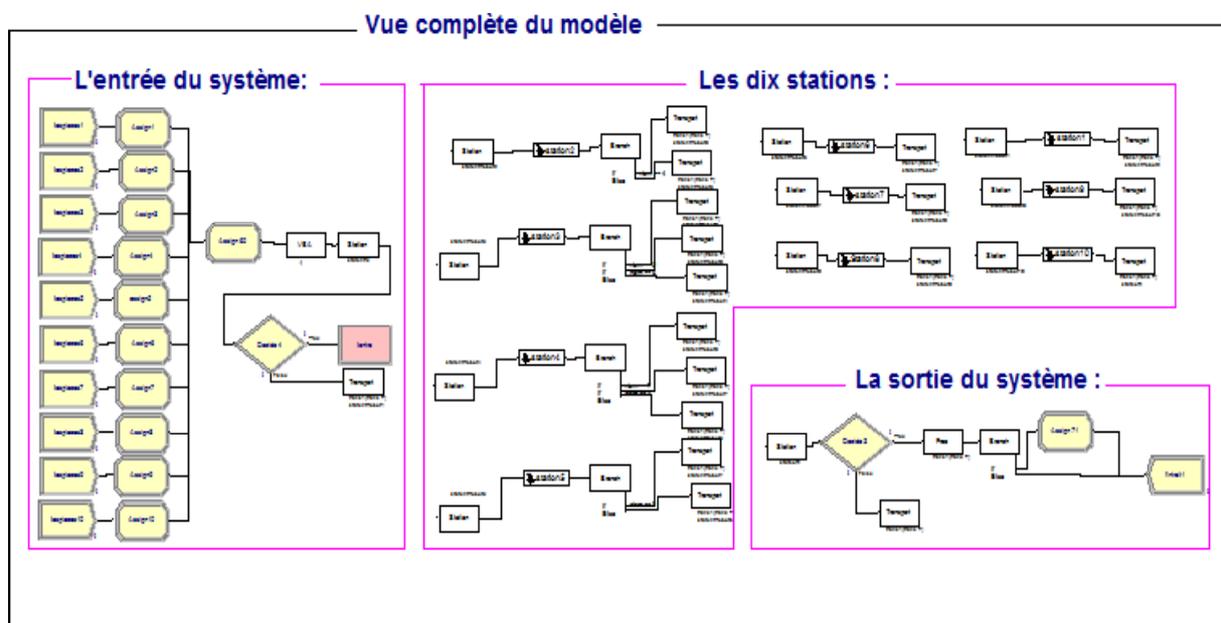


Figure III.2 : Vue complète du modèle

a) Modélisation de l'entrée du système :

L'entrée de la cellule est conçue tel que montré dans la figure III.3

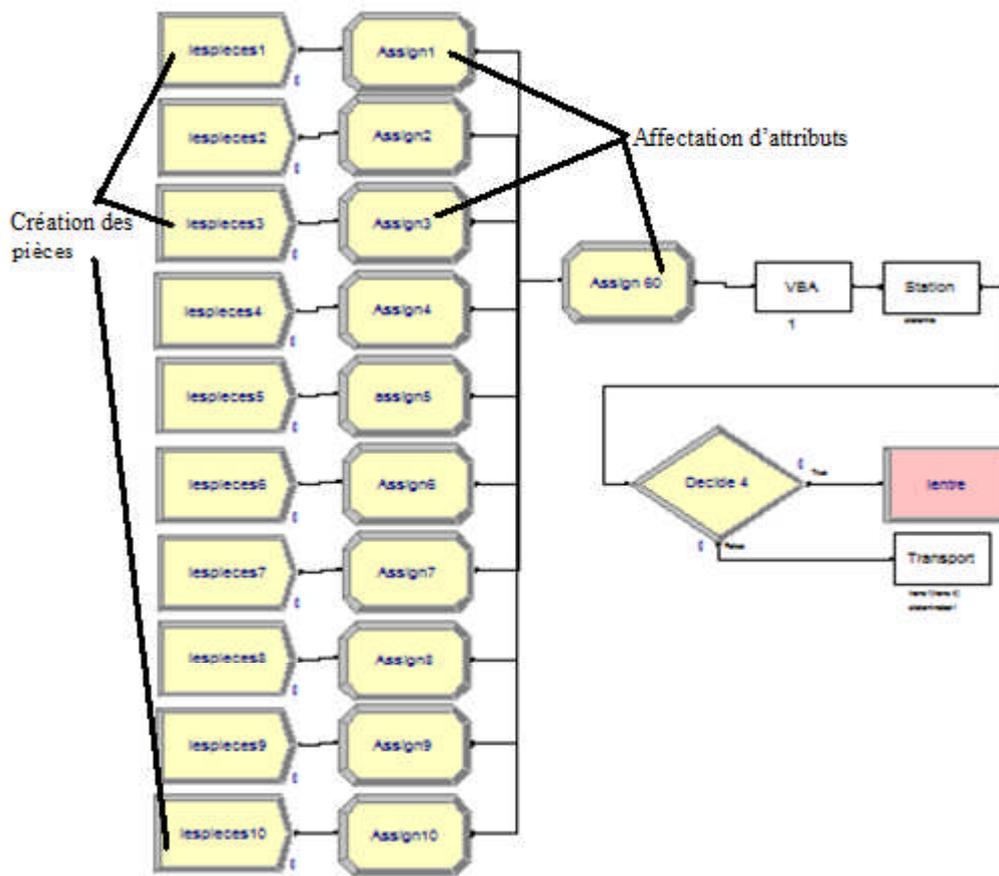


Figure III.3 : Modélisation de l'entrée du système.

On a utilisé dix combinaisons des modules *Create-Assign* afin de créer dix différents types de pièces. Ils représentent nos dix travaux (Jobs) considérés dans cette étude : *Piece1*, *Piece2*, *Piece3* et *Piece4*, *Piece5*, *Piece6*, *Piece7*, *Piece8*, *Piece9* et *Piece10*.

Grâce aux 10 modules *Assign*, on a affecté différents attributs aux entités créées :

- picture : associe une image propre à chaque type de pièce.
- creation : représente le temps de création de l'entité.

Grâce au dernier module *Assign*, on a affecté les attributs suivants à toutes les entités créées.

- Les dix attributs suivants : do1, do2, do3, do4, do5, do6, do7, do8, do9 et do10, représentent les différentes durées opératoires assignées à l'entité.
- Un dernier attribut dtot représente la somme des durées opératoires associées à une entité donnée et qui sera mis à jour à chaque étape d'assemblage.

Une fois l'entité créée, elle passe par un block *VBA* où le code suivant sera exécutée.

- Initialisation de la variable *NBR* (nombre de instances).
- Annulation des attributs do_i pour un rang supérieur au nombre de stations (exemple : si le nombre de stations = 7 alors do_8 , do_9 et do_{10} sont nuls).
- Calcul et affectation de l'attribut *dtot*.

Par la suite l'entité exécute un test dans un block *decide* et si elle vient d'être créée elle sera stockée dans une file d'attente d'entrée nommée *Qentrée* à l'aide de la fonctionnalité *Queue* du module *Leave* où elle reste jusqu'à ce qu'une palette se libère. Dès qu'une palette est libre, elle est réquisitionnée pour transporter la pièce à l'aide de la fonctionnalité *Request transporter* du module *Leave*. L'entité est alors directement mise en mouvement vers la première station à l'aide de la fonctionnalité *transport* du module *Leave*. Dans le cas contraire, ça voudrait dire qu'elle a déjà réquisitionné un transporteur (palette) alors elle sera transportée vers la prochaine station (station 1).

Remarque : Une fois qu'une entité est associée à une palette (transporteur), ARENA leur attribue un seul et unique indice c'est-à-dire qu'elles sont confondues.

b) Modélisation d'une station

La cellule est composée de dix stations, chacune est modélisée par un submodel (figure III.4).

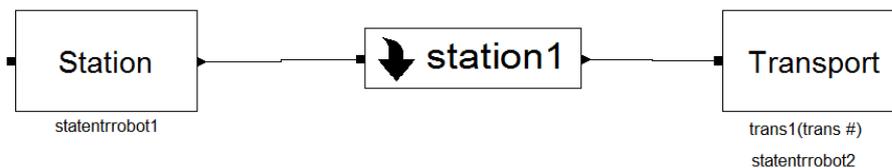


Figure III.4 : Modélisation d'une station par un submodel (station 1).

Le contenu d'un submodel est illustré dans la figure suivante :

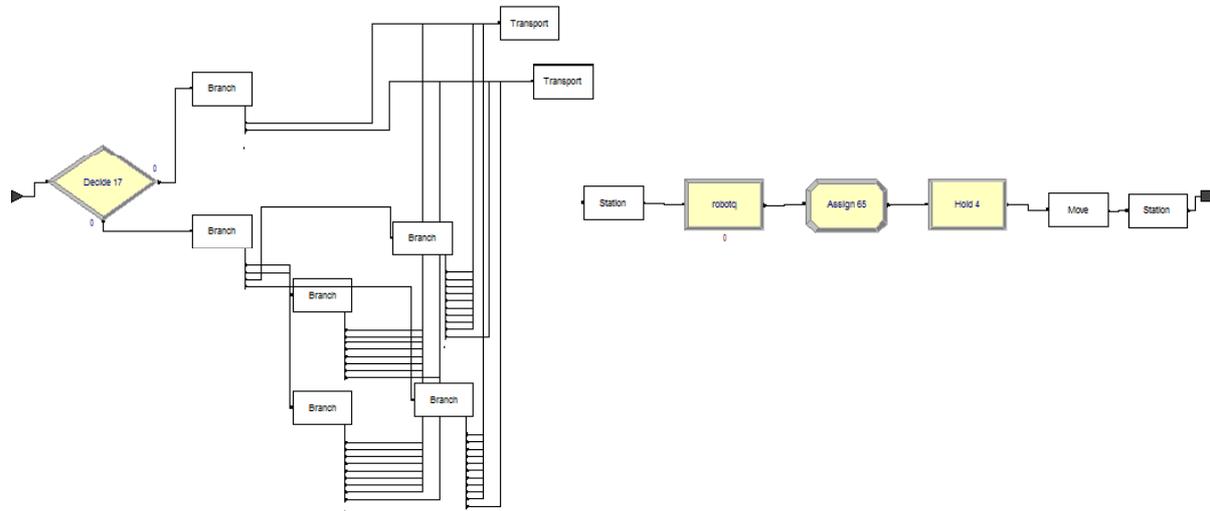


Figure III.5 : contenu d'un submodel (station 4).

Une station est modélisée en trois étapes :

- L'entrée de la station ;
- Le Robot ainsi que les deux files d'attente d'entrée et de sortie (figure III.6) ;
- La sortie de la station (figure III.7).

➤ **Modélisation de l'entrée d'une station:**

Afin de suivre la logique d'un comportement d'une organisation d'atelier Open-Shop et Job-shop une série de tests d'entrée est effectuée à chaque entrée de station dans le but de vérifier quel est l'ordonnancement utilisé (le block *decide*) et si l'entité doit être traitée dans cette station ou pas.

Exemple : station « 1 »

Si choose =1 ou choose = 3 (* Open-Shop ou Flow shop *) **alors**

Si do1 < > 0 et nombre d'entité en attente dans la file de la station « 1 » < FAE **alors**

Transporter l'entité à la file d'attente de la station « 1 » **sinon**

Transporter l'entité à la sortie de la station « 1 »

Fin Si

Sinon Si choose = 2 (* Job shop *) **alors**

Si nbrm = 4 (* 4 stations *) **alors**

Si la pièce est de type 1 et do1 < > 0 et nombre d'entité en attente dans la file de la station « 1 » < FAE **alors** transporter l'entité à la file d'attente de la station « 1 »

Fin Si

Si la pièce est de type 2 et do1 < > 0 et do10=0 et nombre d'entité en attente dans la file de la station « 1 » < FAE **alors** transporter l'entité à la file d'attente de la station « 1 »

Fin Si

⋮

Sinon transporter l'entité à la sortie de la station « 1 »

Fin Si

Si nbrm = 5 (* 5 stations *) **alors**

⋮

etc.

Fin Si.

➤ Modélisation de la station d'assemblage (robot et files d'attente)

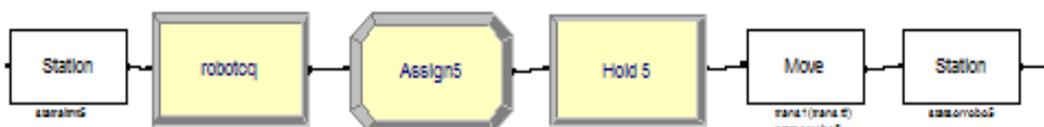


Figure III.6 : Modélisation de la station.

Une fois, l'entité est admise dans la file d'attente elle y reste jusqu'à ce que le robot soit libéré. Selon la règle de priorité appliquée au niveau de la file d'attente, la pièce prioritaire est traitée (*Station, Process*). Dès que le traitement de la pièce est fini, l'attribut de la durée opératoire effectuée est tout de suite assigné à zéro pour marquer le passage de la pièce dans cette station, ainsi qu'une mise à jour de la valeur de *dtot* (*Assign* : $dtot=dtot - doi ; doi=0$). La palette est mise dans une file d'attente de sortie *QSi* (via le module *Hold*)

➤ **Modélisation de la sortie de la station**

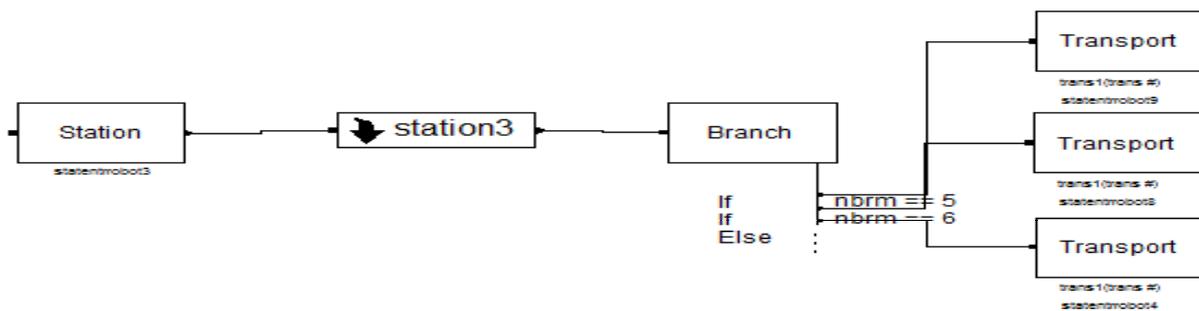


Figure III.7 : Modélisation de la sortie d'une station.

Une fois que l'entité est dans la file d'attente de sortie *QSi*, elle n'en sort que si le point de branchement entre la sortie de la station et le convoyeur est libre sachant que la priorité revient aux pièces sur le convoyeur principal. Si ce point de branchement n'est pas occupé alors la pièce est tout de suite remise sur le convoyeur central sinon elle reste dans la file d'attente de sortie de la station. La prochaine station est choisie selon le nombre de stations arrêté par l'expérience (4,...) à l'aide du block *branch*.

c) **Modélisation de la sortie du système:**

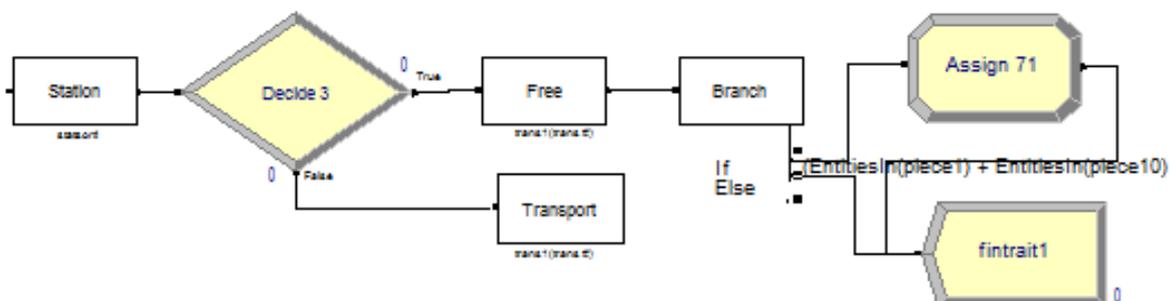


Figure III.8 : Modélisation de la sortie du système.

A chaque fois qu'une palette arrive à la sortie du système, est testé si toutes les opérations d'assemblage sont terminées ($doi=0, i = 1 ; 10$), si c'est le cas la palette est libérée et l'entité est détruite sinon elle est dirigée vers la station1.

Avant la destruction de l'entité, un block *branch* teste si c'est la dernière pièce de la commande ($((EntitiesIn (piece1) ++EntitiesIn (piece10) = ((EntitiesOut (piece1)+ ... + EntitiesOut (piece10) + 1))$), si c'est le cas la variable *CMAXX* (Makespan) est mise à jour à l'aide du block *Assign* ($CMAXX= TNOW$).

III.2.2. Interface d'expérience :

Vu la complexité et la multitude de paramètres considérés, une interface graphique à été réalisée sous VBA (Figure III.9) afin de permettre le paramétrage de l'expérience à simuler (Open-Shop, Job-Shop, Flow-Shop, FIFO, la configuration de la cellule : nombre de stations, distance entre stations, capacité des files d'attente, ...)

Cette interface permet d'interagir avec le modèle en automatisant le changement de variables, d'expressions et les *operands* ^[1] des modules de la manière suivante :

- Déclaration d'un module dans le fichier VBA
- Affectation de ce module au module dans le fichier Arena
- Réalisation des changements sur le module déclaré dans VBA
- Apport de ces changements sur le module déclaré dans Arena

Étape 1 : Déclaration du module

Sous VBA on déclare un élément comme étant du type *module*.

Exemple : les variables : *choose*, *FAE* et l'expression *durée opératoire*

```
Dim mychoose As Module
Dim myFAE As Module
Dim myexpression As Module
Dim index1, index2, index3 As Long
```

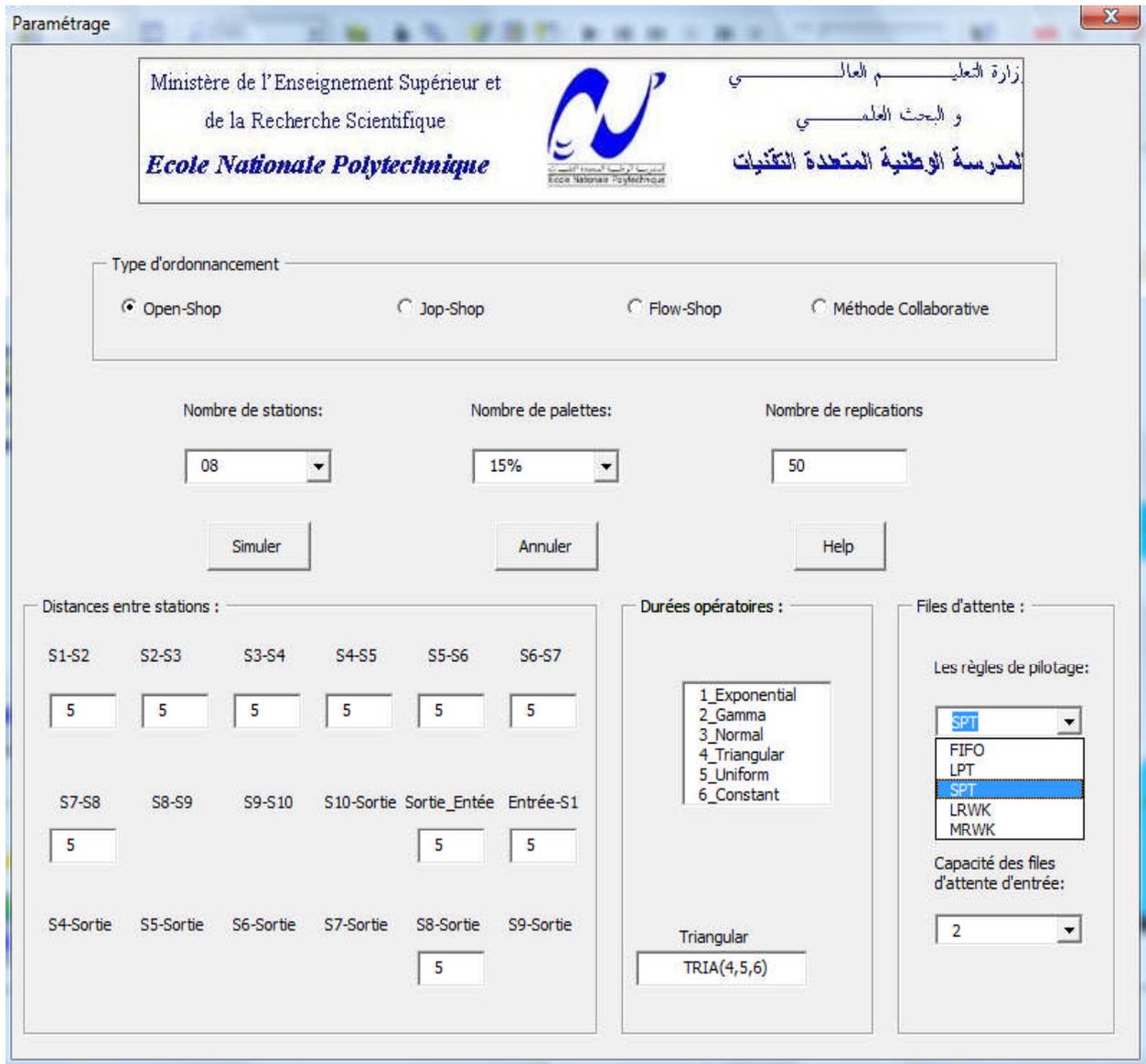


Figure III.9 : Interface d'expérience

Etape 2 : Affectation du module à celui dans le fichier Arena

Chaque objet dans Arena possède un Tag qui lui est assigné par défaut, cependant l'utilisateur peut le modifier (figure III.10).

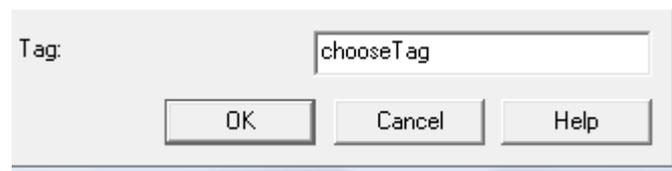


Figure III.10 : Changement du Tag de la variable *choose*

La recherche du module déclaré dans Arena par son Tag et l'affectation de celui déclaré dans VBA à ce dernier ce fait à l'aide de la séquence suivante :

```
'***** la variable choose *****
index1 = m.Modules.Find(smFindTag, "chooseTag")
Set mychoose = m.Modules.Item(index1)

'***** la variable FAE *****
index2 = m.Modules.Find(smFindTag, "FAETag")
Set myFAE = m.Modules.Item(index2)

'***** l'expression durée *****
index3 = m.Modules.Find(smFindTag, "dureeTag")
Set myexpression = m.Modules.Item(index3)
```

Etape 3 : Réalisation des changements sur le module déclaré dans VBA

Exemple : la variable FAE

```
Select Case ComboBox4.value 'ComboBox4 est le menu capacité files d'attente
  Case 0 '02 unités
    myFAE.Data("Initial Value") = 2
  Case 1 '04 unités
    myFAE.Data("Initial Value") = 4
  Case 2 '06 unités
    myFAE.Data("Initial Value") = 6
End Select
```

Etape 4 : Apport des changements sur le module déclaré dans Arena

Les changements effectués sur le module déclaré dans VBA s'opèrent dans celui déclaré dans Arena grâce au mot clé *UpdateShapes*.

```
myFAE.UpdateShapes
```

III.2.3. Interface des résultats :

Une autre interface à été réalisée sous VBA pour nous permettre d'avoir le critère de performance qui nous intéresse dans cette étude à savoir le temps de séjour total d'une commande (*Cmax*) ou le *Makespan*, cette interface (Figure III.11) nous rappelle le niveau pris par chaque paramètre arrêté par l'expérimentateur, ainsi que la moyenne du *Cmax* des expériences réalisées. Elle nous permet aussi soit de continuer la simulation ou de l'arrêter.

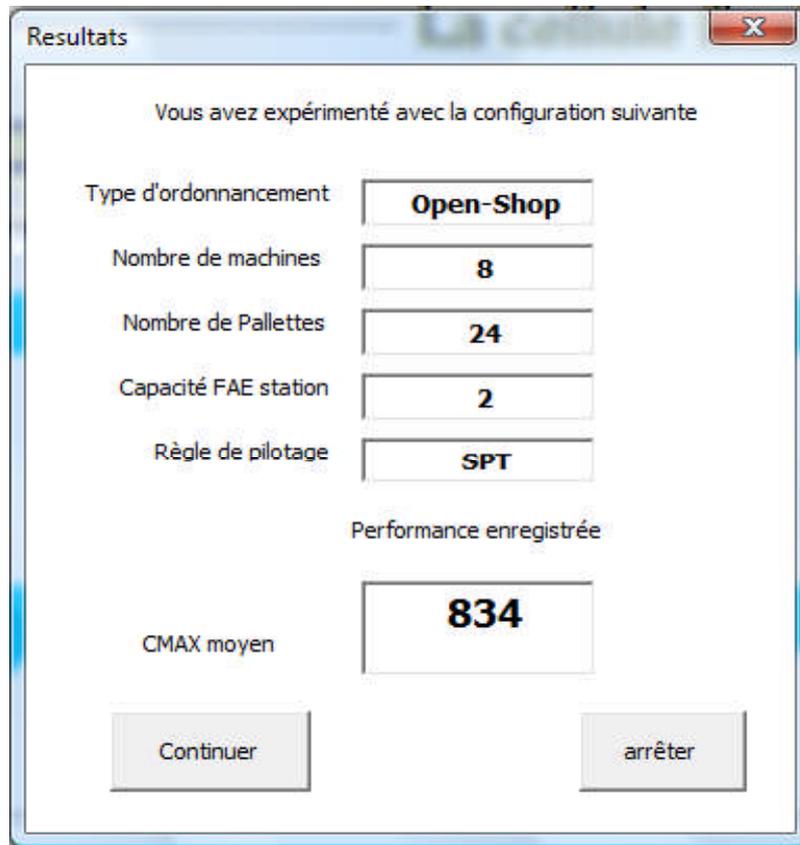


Figure III.11 : Interface des résultats

A la fin de chaque expérience un code VBA (RunEndReplication) s'exécute, il nous permet de :

- Récupérer les valeurs des variables : CMAXX, FAE, choose, nbrm et NBR
- Tester si c'est la dernière expérience pour :
 - Calculer le Cmax moyen
 - Editer les différents champs textes de l'interface

Le code VBA est le suivant :

```
Private Sub ModelLogic_RunEndReplication()  
Dim m As Arena.Model  
Dim s As SIMAN  
Set m = ThisDocument.Model  
Set s = m.SIMAN  
Dim chse, nbrma, fae As Double  
  
cmaxx = cmaxx + s.VariableArrayValue(s.SymbolNumber("CMAXX"))  
chse = s.VariableArrayValue(s.SymbolNumber("choose"))  
fae = s.VariableArrayValue(s.SymbolNumber("FAE"))  
nbrma = s.VariableArrayValue(s.SymbolNumber("nbrm"))  
NBR = s.RunCurrentReplication  
  
m.QuietMode = True  
  
If s.RunMaximumReplications = s.RunCurrentReplication Then  
cmaxx = (cmaxx / NBR)  
Resultats.TextBox2.value = nbrma  
Resultats.TextBox3.value = s.TransporterNumberOfUnits(1)  
Resultats.TextBox4.value = fae  
Resultats.TextBox5.value = Fix(cmaxx)  
Resultats.Show  
End If  
  
End Sub
```

III.2.4. Modélisation dynamique de l'atelier d'assemblage

Le schéma graphique de l'atelier par ARENA est présenté dans la figure III.12

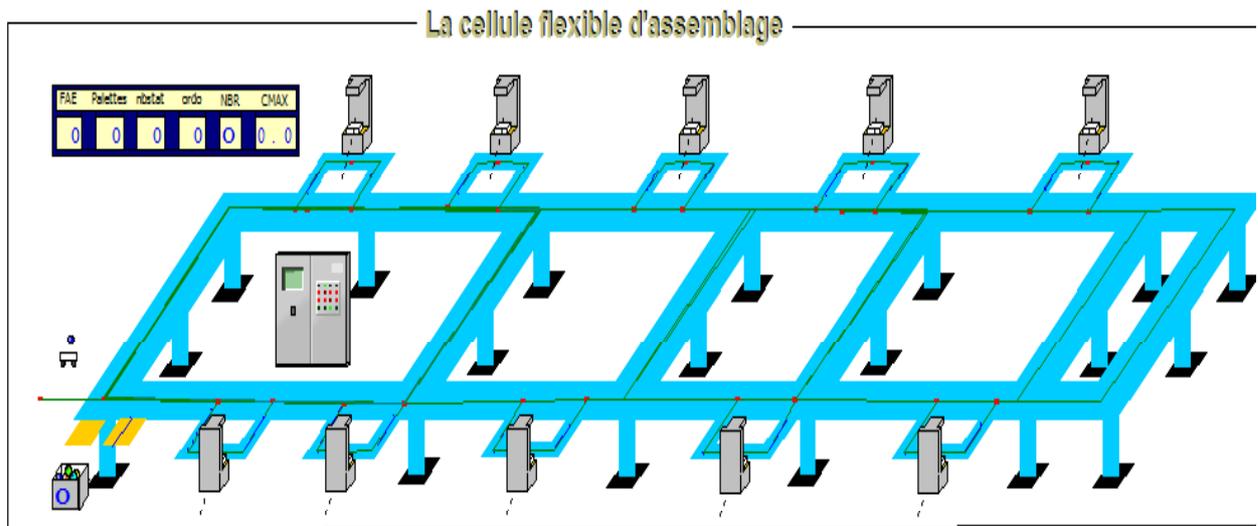


Figure III.12 : La représentation graphique de la cellule

On a représenté les différents points d'entrée et de sortie des stations à l'aide de la fonctionnalité *AnimateTransfer/Station* du logiciel Arena, comme montré dans la figure.

On a représenté les différentes files d'attente d'entrée (FAEi) des stations et les différentes files d'attente de sortie des stations (FASi) par *Animate/Queue* (figure III.13).



Figure III.13: a. Station, b. file d'attente d'entrée des stations, c. File d'attente de sortie

Les palettes sont représentées par des transporteurs générés dans le module *Transporter* (Figure III.14).

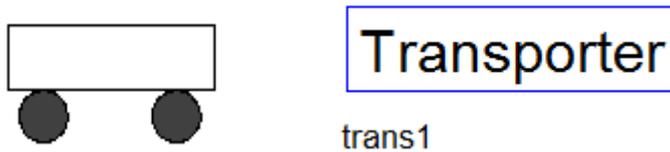


Figure III.14 : modélisation des Palettes par des transporteurs

La figure suivante représente une image du système en pleine évolution. On y voit des palettes sur le convoyeur, des pièces dans les files d'attente ainsi que des robots en plein traitement.

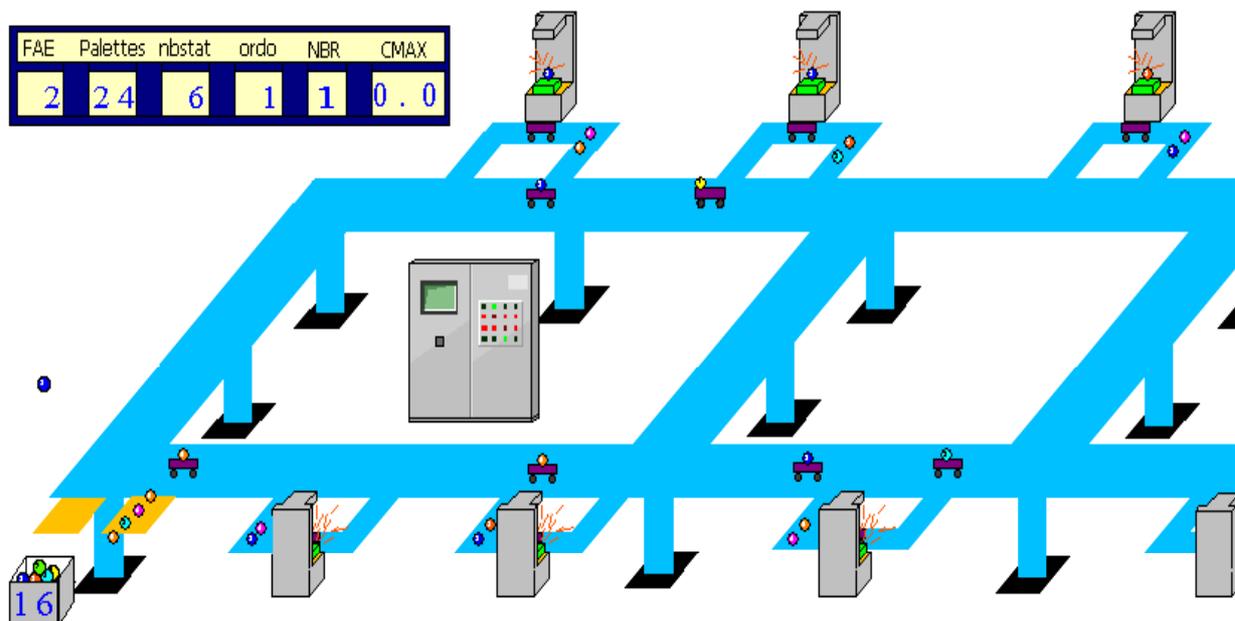


Figure III.15: le système en pleine évolution (6stations)

III.3. Vérification et validation du modèle

On a soumis le modèle obtenu à une série de tests, qui nous ont permis de vérifier son bon fonctionnement et de le valider par la suite.

- On a suivi l'évolution du système avec une seule palette pour s'assurer de son cheminement à travers la cellule ;

- On a augmenté la taille de la file d'attente de l'entrée des stations indéfiniment l'une après l'autre pour savoir si elle absorbera toutes les palettes circulantes ;
- On a éliminé les durées opératoires par rapport à une telle station, de toutes les pièces, l'une après l'autre pour s'assurer de son inactivité pendant toute la simulation ;
- On a mis des files d'attente avec des capacités nulles pour s'assurer du respect de la condition « file d'attente non saturée » ;
- On a défini des séquences de travail propres à chaque type de pièces pour valider le Job Shop.

On a également utilisé :

- Des techniques d'animation du modèle montrant les entités en déplacement conformément aux règles de fonctionnement : ainsi, grâce à l'animation mise en place, on a pu confirmer que la circulation des entités est cohérente avec les règles mises en place ;
- Des comparaisons avec les résultats obtenus dans (Ghazi, 2007);
- L'utilisation des fichiers "sortie" pour analyser les données recueillies au cours de la simulation.

Par ailleurs, Arena nous permet d'afficher en temps réel durant la simulation, la valeur des variables calculés au niveau des blocks *VBA*, ce qui nous permet de garder une traçabilité du fonctionnement du système et de le comparer au fonctionnement souhaité. Nous avons donc pu suivre le fonctionnement du système afin de s'assurer que les conditions définies dans les blocks *VBA* étaient correctement appliquées.

IV. Conclusion.

Dans le cadre de ce chapitre, on a réalisé les six premières étapes du processus de simulation telles présentées par (Pritsker, 1986).

On a commencé ce chapitre par un rappel de la problématique étudiée, afin de structurer la modélisation autour des axes définis par notre approche. La création d'un modèle fidèle au système étudié et sa transcription informatique sont les deux étapes les plus délicates d'un processus de simulation et c'est auxquelles une grande partie du temps a été consacrée. On a conçu un modèle de la cellule étudiée. Par la suite, on l'a validé par un ensemble de tests (d'abord simples par la suite extrêmes). Une fois le modèle validé, nous procéderons à

l'application des trois étapes restantes du processus de simulation dans le cadre du prochain chapitre. La prochaine étape est la planification stratégique et tactique de la simulation.

Chapitre IV

SIMULATION ET TESTS DES DIFFERENTES STRATEGIES D'ORDONNANCEMENT PROPOSEES.

*“On fait la science avec de faits, comme on fait
une maison avec des pierres : mais une
accumulation de fait n'est pas plus
une science qu'un tas de pierres
n'est une maison”*

[Henri Poincaré]

Chapitre IV : Simulation et tests des différentes stratégies d'ordonnancement proposées.

- I. Introduction
- II. La planification stratégique et tactique de la simulation
- III. Les indicateurs de performance à analyser
- IV. L'analyse et l'interprétation des résultats
 - IV.1. L'impact des ordonnancements Open-Shop et Job-Shop sur le *C_{max}*
 - IV.2. L'impact des règles de priorité au niveau des stations dans le cas Open-Shop
 - IV.3. L'impact de l'approche collaborative sur le *C_{max}*.
- V. Conclusion

I. Introduction :

Dans ce chapitre, on met en œuvre les étapes restantes du processus de simulation, telles que définies par Pritsker (Pritsker, 1986), c'est-à-dire la planification stratégique et tactique de la simulation, son exécution selon l'étape précédente et enfin l'analyse et l'interprétation des résultats.

Pour ce faire, la première section de ce chapitre est consacrée à la planification des expériences. La seconde section est consacrée à l'analyse et interprétation des résultats. On analysera les temps de séjour (d'assemblage, C_{max}) d'une commande de pièces.

La troisième et dernière section est une conclusion sur les principaux résultats de ce chapitre.

II. La planification stratégique et tactique de la simulation :

Selon l'objectif arrêté de l'étude, l'approche n'est pas toujours simple à mettre en œuvre car les paramètres à considérer lors de l'étude de la cellule d'assemblage, sont très nombreux et chacun d'eux peut avoir son influence sur la performance globale de la cellule. Ceci constitue une situation combinatoire qui pourrait «exploser» exponentiellement.

Sachant qu'un temps de séjours d'une pièce dans la cellule est composé des temps d'attente dans les files d'attente, des durées opératoires sur les différents postes de travail ainsi que des temps de transfert entre les stations (les temps de chargement et déchargement sont considérés nuls), sa minimisation fera intervenir plusieurs paramètres. Parmi ces paramètres la capacité des FAE (un temps d'attente dans une FAE peut être beaucoup plus faible qu'un tour complet sur le convoyeur central).

Dans le cadre de cette étude, on a considéré les paramètres suivants :

"la longueur du convoyeur central, le nombre de stations, la capacité des files d'attente d'entrée, le nombre de palettes, les règles de priorité, le type d'ordonnement et dans un premier temps la taille de la commande."

La planification des expériences s'articulera autour des axes :

- Impact d'un ordonnancement Job-Shop et d'un ordonnancement Open-Shop;
- Impact des règles de priorité dans le cas Open-Shop ;
- Apport de l'approche distribuée ;
- Impact de la structure de la cellule (4 stations,, 10 stations).

Une fois démontré que l'apport d'une combinaison plans non linéaire-pilotage FIFO est peu significatif, on s'intéressera à l'ordonnement par règle de priorité dans le but de diminuer les temps d'attente et de transfert simultanément, le tout en vérifiant la stabilité des résultats sous des structures d'expérimentation différentes (4 stations,

Pour une couverture suffisante du comportement du système étudié, on a défini une planification adaptée des expériences à mener. Ceci est détaillé dans le tableau ci-dessous :

Paramètres	Ordonnancement	règles	convoyeur	stations	Files-Attente	palettes	commande
Niveaux	1. FS 2. OS 3. JS 4. Collaboratif (à 4 stations)	1. FIFO 2. LPT 3. SPT 4. LRWK 5. MRWK	I. 5 II. 7 III. 10	I. 4 II. 5 III. 6 IV. 7 V. 8 VI. 9 VII. 10	I. 2 II. 4 III. 6	I. 10% II. 15% III. 20% IV. 25%	I. 60 II. 80 III. 100 (à 4 stations)

Tableau IV.1: Planification des d'expériences.

En tout, on a effectué **2700** expériences nettes (**135000** instances).

III. Les indicateurs de performance à analyser :

Pendant la phase d'analyse des résultats, pour chaque expérience, on calculera les taux d'amélioration entre les résultats obtenus en (FS, FIFO) et ceux obtenus de l'OS et JS (ainsi que *l'approche collaborative* à 4 stations), c'est la première partie. Dans la seconde partie, on calculera les taux d'amélioration entre les résultats obtenus en (OS, FIFO) et ceux obtenus des autres règles (toujours OS). Les temps étudiés sont les temps de séjours d'une commande de pièces dans la cellule flexible d'assemblage.

Dans le premier cas, on calcule l'amélioration correspondant à une réduction du temps de séjour d'une commande en suivant la première partie. Les taux d'amélioration sont calculés de la manière suivante:

$$\tau_{TSCOrdo.} = \left(\frac{T_{SCFS} - T_{SCOS}}{T_{SCFS}} \right) \times 100^{(cas1)}, \quad \tau_{TSCOrdo.} = \left(\frac{T_{SCFS} - T_{SCJS}}{T_{SCFS}} \right) \times 100^{(cas2)}$$

Où :

T_{SCFS} : est le temps de séjour d'une commande sous un ordonnancement FS.

T_{SCOS} : est le temps de séjour d'une commande sous un ordonnancement OS.

T_{SCJS} : est le temps de séjour d'une commande sous un ordonnancement JS.

Dans le deuxième cas, on calcule l'amélioration correspondant à une réduction du temps de séjour d'une commande en suivant la deuxième partie. Les taux d'amélioration sont calculés de la manière suivante:

$$\tau_{TSCOrdo.} = \left(\frac{T_{SCOSFIFO} - T_{SCOSRègle}}{T_{SCOSFIFO}} \right) \times 100$$

Où :

$T_{SCOSFIFO}$: est le temps de séjour d'une commande sous un ordonnancement OS, FIFO.

$T_{SCOSRègle}$: est le temps de séjour d'une commande sous un ordonnancement OS, règle de priorité utilisée.

IV. L'analyse et l'interprétation des résultats :

IV.1. L'impact des ordonnancements Open-Shop et Job-Shop sur la cellule flexible

En respectant la planification des expériences élaborée précédemment (grâce à l'étape planification stratégique et tactique), on a mis en œuvre et expérimenté les nombreuses et différentes configurations et relevé les résultats qu'on a analysés et interprétés par la suite.

Dans cette section précise, l'analyse des résultats concerne l'impact des ordonnancements Open-Shop et Job-Shop sur les performances (C_{max}) de la cellule flexible d'assemblage étudiée. En fait, on va comparer les performances de la cellule d'assemblage d'abord sous une organisation Job-Shop, par la suite sous une organisation Open-Shop tout en vérifiant les résultats avancés par l'étude effectuée par Finke et Dupont. On applique à chacune des deux organisations un ordonnancement par la règle de priorité FIFO (First In First Out).

Finke et Dupont ont conclu que les performances d'un système d'assemblage cyclique sont conditionnées par le robot goulot et que son pilotage par des plans non linéaires (Job-shop) n'apportait aucune amélioration aux performances du système, mais que l'intérêt est dans la manipulation des plans d'assemblage ouverts d'où l'acquisition d'une flexibilité dans l'ordre de passage des tâches (l'Open-shop), où chaque ordre des opérations d'assemblage est possible. Ils ont montré que l'apport d'une combinaison plans non linéaire-pilotage FIFO à la performance du système est peu significatif (inférieur à 5%).

Pour rappel, le nombre d'expériences menées dans cette partie d'analyse est de 1080 (résultats), chaque résultat est issu d'une simulation de 50 instances exécutées par le logiciel ARENA12. Les paramètres de variation sont les suivant :

- Nombre de palettes : 10%, 15%, 20% et 25% de la taille de la commande (exemple, pour une commande de 80 pièces, $15\% \equiv 80 \times 0.15 = 12$ palettes);
- Nombre de stations : 4, 5, 6, 7, 8, 9 et 10 ;
- Capacité des files d'attente d'entrée : 2, 4 et 6 entités ;

- Distance entre les stations : 5, 7 et 10 (l'unité est la palette) ;
- Et dans le cas à 4 stations la taille de la commande : 60, 80 et 100 entités (pièces à assembler).

Une fois montré que l'impact de la taille de la commande sur les performances de la cellule est peu signifiant, la taille de la commande sera 20 fois le nombre de stations i.e. à 6 stations la taille de la commande sera $20 \times 6 = 120$ pièces.

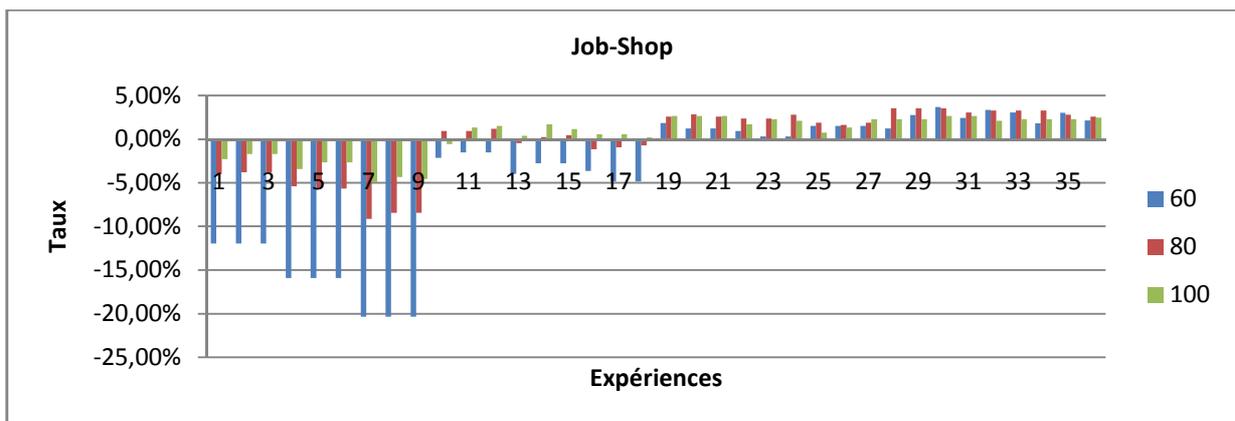
On va donc analyser la performance globale (*Cmax*) de la cellule induite par les différents ordonnancements. Pour ce faire, nous allons interpréter les temps de séjour dans la cellule d'un ensemble de pièces correspondant à une commande.

Analyse des temps d'assemblage d'une commande de pièces, minimisation du *Cmax* :

L'analyse des résultats s'est faite par la lecture des graphes en considérant les paramètres de variation arrêtés précédemment.

a). **Analyse préliminaire** : Impact de la taille de la commande

Pour cela, nous avons considéré la cellule flexible à 4 stations et nous avons relevé les résultats suivants :



a)



b)

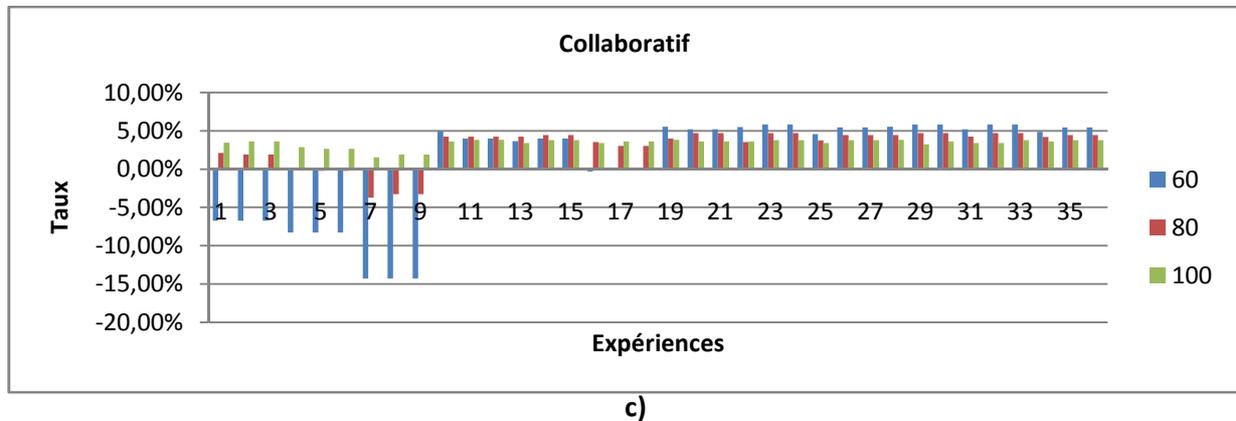


Figure VI.1 : Impact de la taille de commande sur le C_{max} .

D'après les résultats de la simulation (les graphes ci-dessus), l'impact de la taille de la commande sur la performance de la cellule d'assemblage est peu significatif (inférieur à 6%), et ce pour les trois types d'ordonnancement.

En effet, à part l'augmentation du nombre des cas d'amélioration de la performance (toujours peu significative), l'augmentation de la taille de la commande fait diminuer même cette dernière.

A titre d'exemple, pour l'ordonnancement distribué, les cas d'amélioration de la performance (C_{max}) est de l'ordre de 67% pour une commande de 60 pièces et il l'est à 100% pour une commande de 100 pièces. Mais c'est dans ce dernier cas que le maximum des taux d'amélioration 3.81% est inférieur à celui où la commande est de 60 pièces 5.83% (le maximum).

Dans les trois cas de figure (60, 80 et 100), c'est le Job-Shop qui présente les performances les plus dégradées -20.36% (le pire) pour une commande de 60 pièces. Le comportement de l'Open-Shop est similaire à celui de l'approche distribuée même pour le nombre de cas d'amélioration de la performance (exemple : pour une commande de 80 pièces le nombre de cas d'amélioration est de 83% pour l'approche distribuée et il est de 86% pour l'Open-Shop).

Du fait de l'impact quasiment inexistant de la taille de commande sur le C_{max} , nous ne nous intéresserons par la suite, pour l'étude des impacts des différents ordonnancements, qu'à la taille des commandes vérifiant la relation : $\text{taille commande} = 20 \times \text{nombre_stations}$.

Remarque :

Le chiffre 20 est choisi d'une manière arbitraire mais il respecte l'ordre de grandeur de la taille de commande (suivant le nombre de stations).

b). Impact des différents ordonnancements sur le C_{max} :

Taille de la commande = $20 \times \text{nombre_stations}$

On va analyser et comparer en même temps : l'impact des différents paramètres sur les performances enregistrées pour chaque type d'ordonnement, et l'impact de chaque ordonnancement sur la performance de la cellule flexible.

Remarquons seulement que si on éclate les expériences sur les niveaux d'un paramètre de variation alors, la somme des expériences suivant ces différents paramètres est le nombre d'expériences totales. Ainsi, une fois l'impact d'un paramètre sur les performances d'un ordonnancement expliqué, l'impact d'un tel ordonnancement (tous niveaux confondus) sur la cellule flexible sera le même pour les autres paramètres (avec le même nombre de stations).

Cas1 : 4 stations

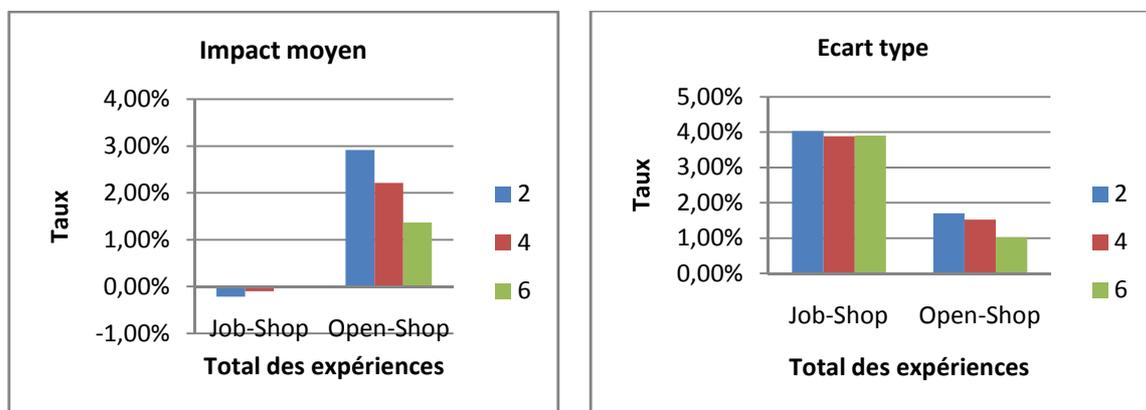
Impact de la capacité des files d'attente sur les performances des ordonnancements:

Figure VI.2 : Impact de la capacité des files d'attente sur le C_{max} .

Remarquons que pour l'ordonnement Open-Shop la moyenne des taux d'amélioration est de l'ordre de **2.92%** avec un écart type de **1.70%** obtenu dans le cas où les capacités des files d'attente est **2**. Le Job-Shop quant à lui, présente des dégradations de performance en moyenne de **- 0.02%** avec un écart type de **3.90%** (capacité des files d'attente 6 entités). Pour l'ordonnement Job-Shop, nous remarquons une amélioration des performances (**C_{max}**) proportionnellement aux capacités des files d'attente, mais cet impact reste très faible. Au contraire, pour l'open-Shop, les performances se dégradent (mais peu significantes).

Ce comportement est dû à la nature même de l'Open-Shop qui, si une pièce arrive à une station elle y accède, pourvu qu'elle a une tâche avec une place dans la file d'attente d'entrée, ce qui fait que les stations suivantes sont moins saturées que les premières, d'où le déséquilibre. Le Job-Shop présente un comportement contraire à celui de l'Open-Shop concernant performances versus capacités des files d'attente. Ceci s'explique par le fait que le

Job-Shop distribue les pièces d'une manière équilibrée (grâce à la gamme opératoire définie avant la résolution du problème d'ordonnancement, *Line-Balancing*).

Nous déduisons de ce qui précède que pour chaque type d'ordonnancement l'impact des capacités des files d'attente sur le Cmax est quasiment inexistant. Ainsi, nous déduisons aussi que l'impact des différents ordonnancements sur le Cmax demeure peu significatif (ne dépasse pas 5%).

Impact de la distance entre stations sur les performances des ordonnancements:

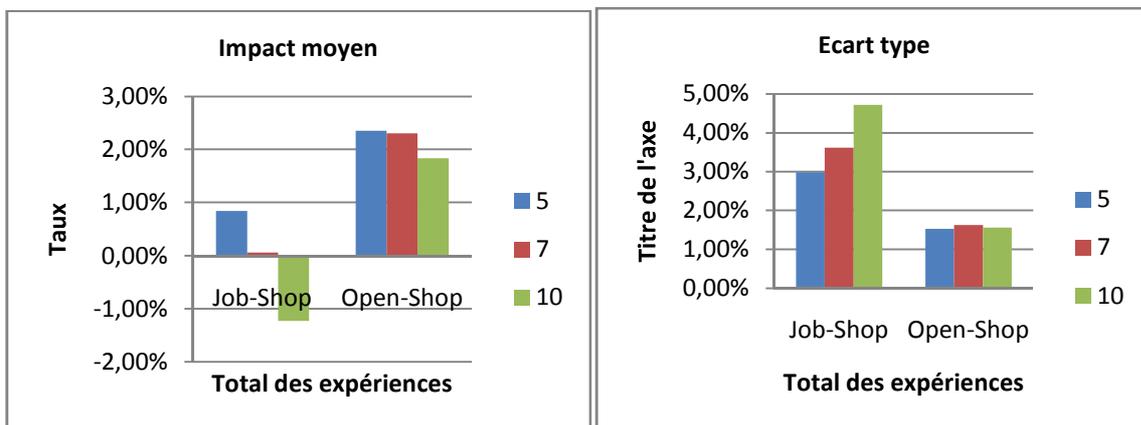


Figure VI.3 : Impact de la longueur du convoyeur sur le Cmax.

Remarquons que pour l'ordonnancement Open-Shop que la moyenne des taux d'amélioration est de l'ordre de **2.35%** avec un écart type de **1.53%** obtenu dans le cas où la distance entre stations égal à 5 palettes également. Le Job-Shop quant à lui présente une performance moyenne de **0.84%** avec un écart type de **2.98%** (distance entre stations est de 5 palettes).

On remarque qu'en moyenne c'est l'Open-shop qui l'emporte sur le Job-Shop. Ceci est dû à la flexibilité dans l'ordre de passage des tâches (où chaque ordre des opérations d'assemblage est possible). L'Open-Shop bénéficie de la flexibilité dans l'ordre de passage des tâches mais pas le Job-Shop. Le Job-Shop présente des contraintes potentielles (de précédence) ce qui élimine la flexibilité dans l'ordre de passage des tâches.

Pour les deux types d'ordonnancement, les performances diminuent avec l'augmentation de la longueur du convoyeur. Ceci s'explique par le fait d'augmentation des temps de transport des pièces d'une machine à une autre. Car rappelons le, le temps de séjour total d'une commande n'est que la somme des temps de transport entre stations, les temps d'attente dans les files d'attente et les temps de traitement.

Nous déduisons de ce qui précède que pour chaque type d'ordonnancement l'impact des distances entre stations (longueur du convoyeur) sur le Cmax demeure peu significatif.

Impact du nombre de palettes sur les performances des ordonnancements:

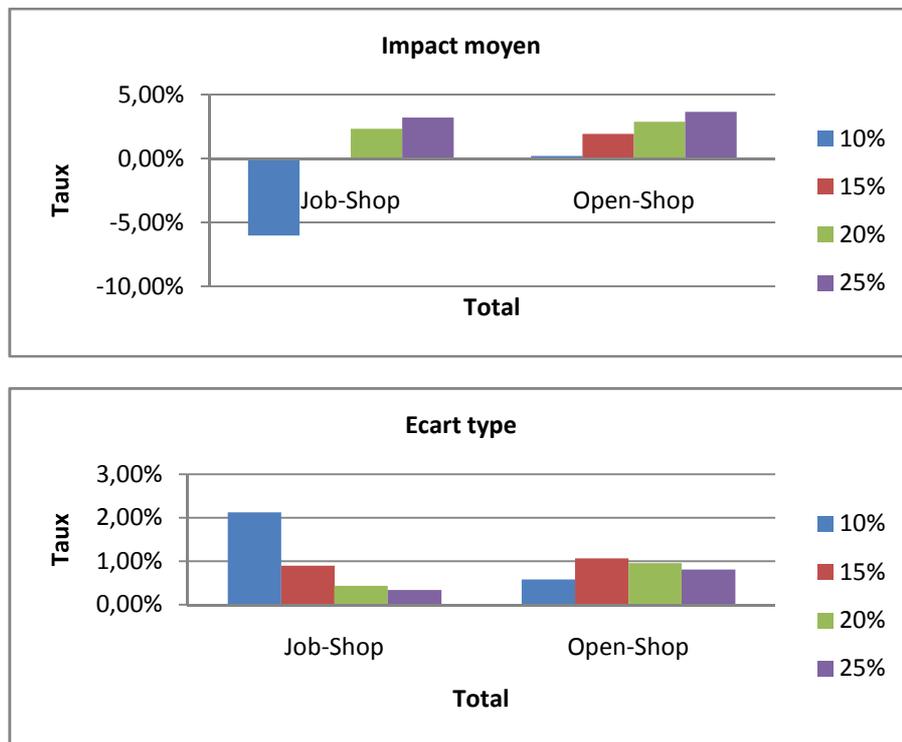


Figure VI.4 : Impact du nombre de palettes sur le Cmax.

D'après les graphes ci-dessus, pour l'ordonnancement Open-Shop, la moyenne des taux d'amélioration est de l'ordre de **3.65%** avec un écart type de **0.81%** obtenu dans le cas où le nombre de palette est de 25% également. Le Job-Shop quant à lui présente une performance moyenne (meilleure que les cas précédents) de **3.21%** avec un écart type de **0.34%** (nombre de palette 25%).

On remarque qu'en moyenne c'est l'Open-shop qui l'emporte sur le Job-Shop (comme le cas précédent).

Pour les deux types d'ordonnancement, les performances augmentent avec l'augmentation du nombre de palettes dans la cellule. Ceci s'explique par le fait que les performances d'une cellule flexible d'assemblage s'améliore en continue jusqu'au niveau optimum défini par Finke et Dupont (on est encore dans l'intervalle appelé *steep descent* par les deux auteurs).

Nous déduisons de ce qui précède que pour chaque type d'ordonnancement l'impact du nombre de palettes (dans notre cas) sur le Cmax demeure peu significatif.

Impact des différentes organisations sur le C_{max} :

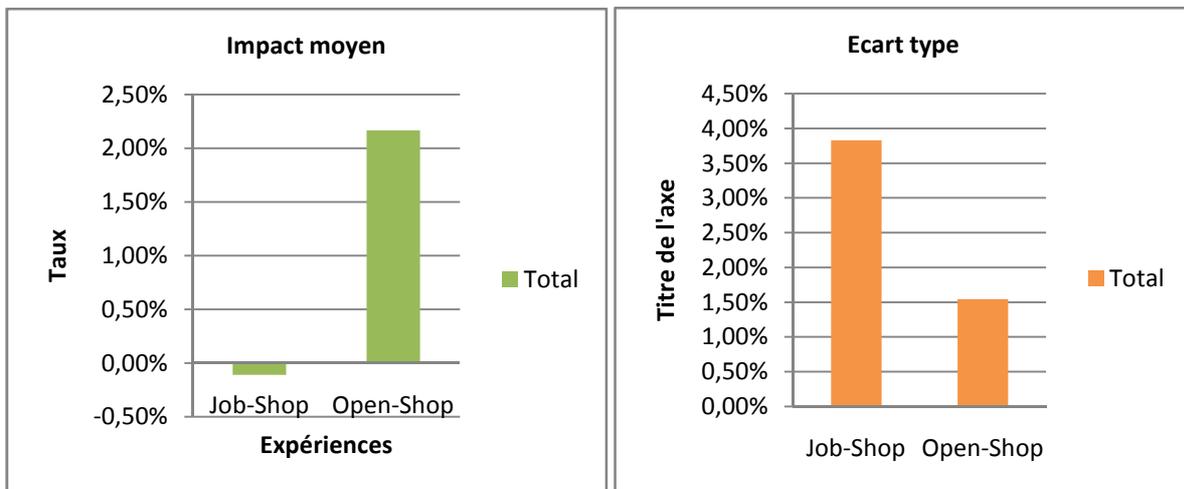


Figure VI.5 : Impact des différentes organisations sur le C_{max} .

En conclusion, à 4 stations, le pilotage par des plans non linéaires (Job-shop) n'apporte aucune amélioration aux performances du système (- 0.11% ± 3.83%), mais l'intérêt est dans la manipulation des plans d'assemblage ouverts (Open-Shop, 2.17% ± 1.54%). Toutefois, son impact demeure peu significatif (< 5%, Finke et Dupont).

On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible, ce qui signifie robustesse et stabilité des résultats

Cas2 : 5 stations

Dans ce cas, l'approche collaborative n'est pas implémentée, elle peut être un sujet d'un projet de fin d'études pour l'année prochaine.

Impact de la capacité des files d'attente sur les performances des ordonnancements:

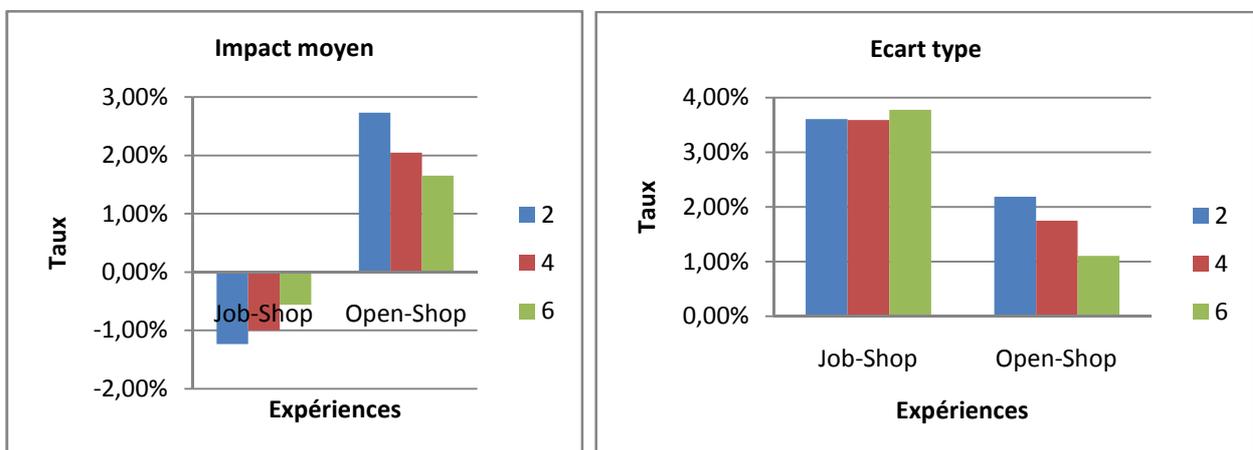


Figure III.6 : Impact de la capacité des files d'attente sur le C_{max} .

On remarque que c'est exactement le même comportement que dans le cas à 4 stations. En effet, l'impact des files d'attentes sur les performances de l'Open-Shop et du Job-Shop est insignifiant. Performance ne dépassant pas les 3% et un écart type de l'ordre de 2% (cas de l'Open-Shop, capacité des files d'attente est 2). Le Job-Shop a enregistré des performances négatives (-0.5% ± 1.1%).

Impact de la distance entre stations sur les performances des ordonnancements:

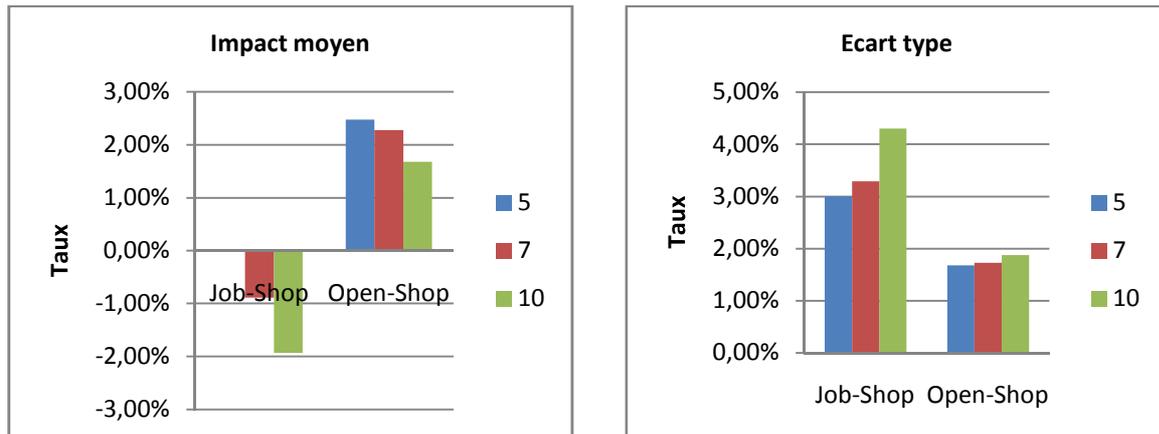


Figure VI.7 : Impact de la longueur du convoyeur sur le Cmax.

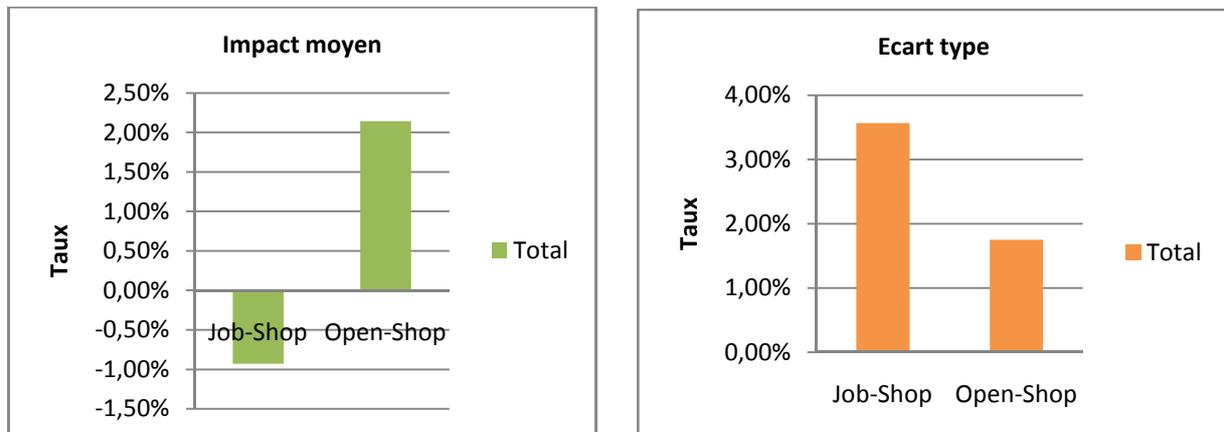
Même conclusion que le cas à 4 stations, même comportement des performances et l'impact de la longueur totale du convoyeur reste insignifiante. (< 3.0% ± 1.5%, Open-Shop, distance 5 palettes).

Impact du nombre de palettes sur les performances des ordonnancements:



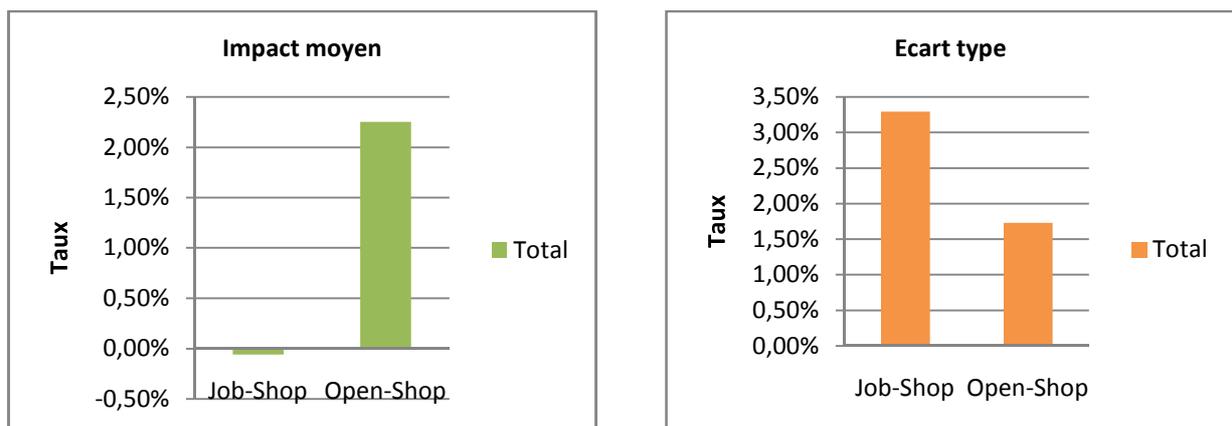
Figure VI.8 : Impact du nombre de palettes sur le Cmax.

Même conclusion que dans le cas à 4 stations, le comportement des performances est le même et l'impact du nombre de palettes (avec les niveaux arrêtés dans notre plan d'expériences) demeure peu significatif (< 4% ± 1%).

Impact des différentes organisations sur le C_{max} :Figure VI.9 : Impact des différentes organisations sur le C_{max} .

En conclusion (à l'instar du cas à 4 stations), à 5 stations, le pilotage par des plans non linéaires (Job-shop) n'apporte aucune amélioration aux performances du système (- 1,0% ± 3,5%), toutefois l'intérêt demeure dans la manipulation des plans d'assemblage ouverts (Open-Shop, 2,15% ± 1,75%). Toutefois, son impact demeure peu significatif (< 5%, Finke et Dupont). On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible, ce qui signifie robustesse et stabilité des résultats.

Cas3 : 6 stations

Figure VI.10 : Impact des différentes organisations sur le C_{max} .

Cas4 : 7 stations

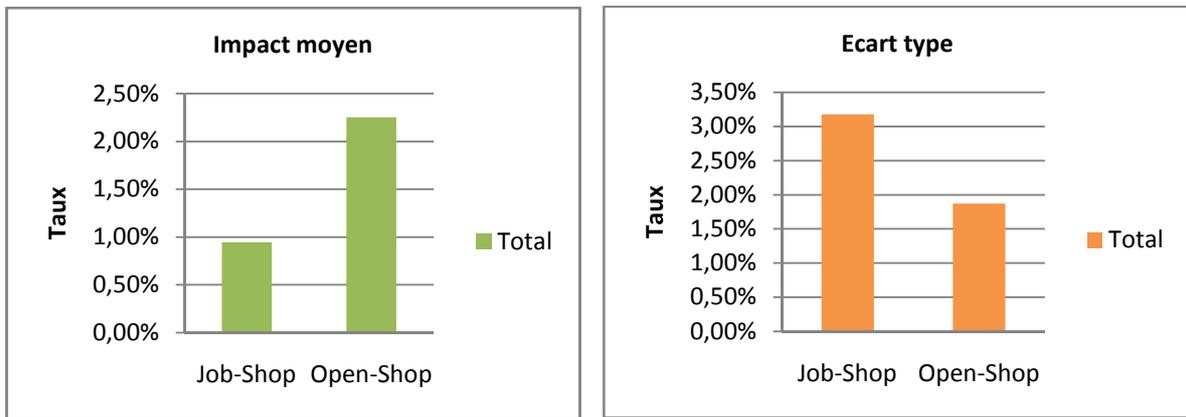


Figure VI.11 : Impact des différentes organisations sur le C_{max} .

Cas5 : 8 stations

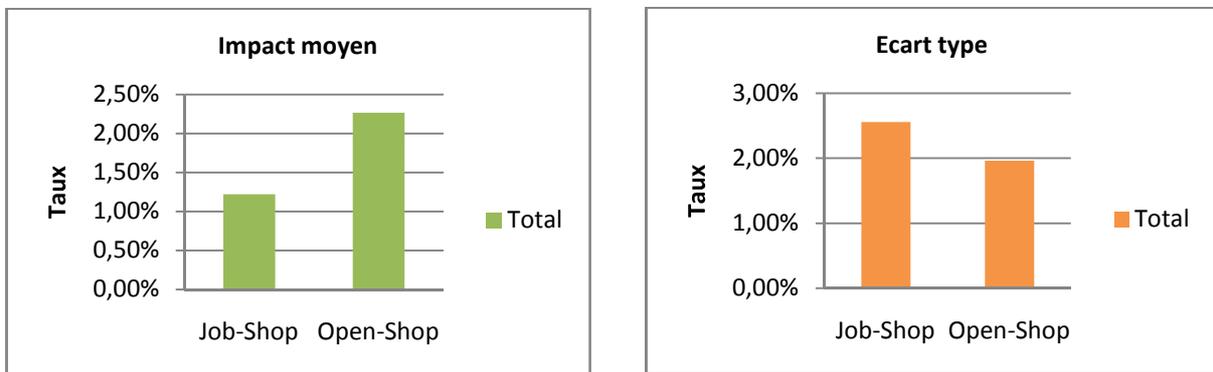


Figure VI.12 : Impact des différentes organisations sur le C_{max} .

Cas6 : 9 stations



Figure VI.13 : Impact des différentes organisations sur le C_{max} .

Cas7 : 10 stations

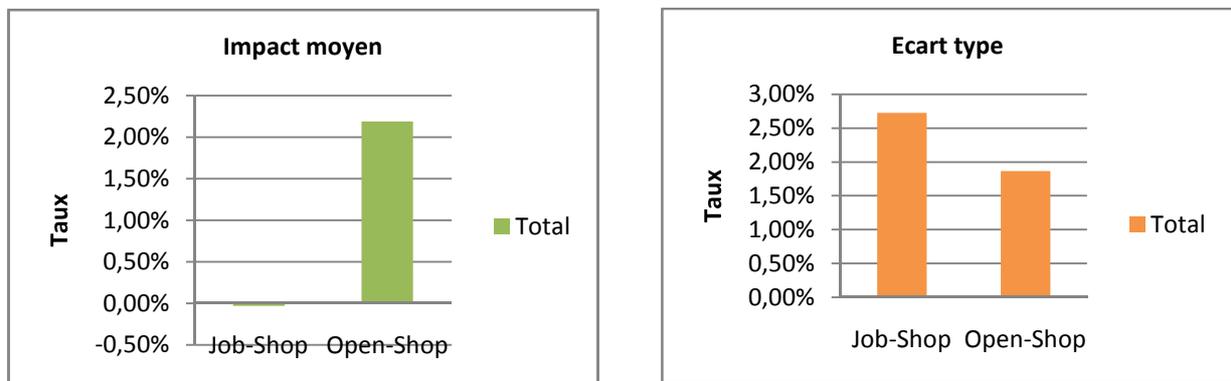


Figure VI.14 : Impact des différentes organisations sur le C_{max} .

En conclusion (les cas ci-dessus), l'impact des différents paramètres ainsi que les différents ordonnancements sur le C_{max} est insignifiant. Le pilotage par des plans non linéaires (Job-shop) n'apporte aucune amélioration aux performances du système ($0\% \pm 3\%$), l'intérêt est toujours dans la manipulation des plans d'assemblage ouverts (Open-Shop, $2\% \pm 2\%$), mais son impact demeure peu significatif ($< 5\%$, Finke et Dupont). On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible, ce qui signifie robustesse et stabilité des résultats indépendamment des paramètres et du nombre de stations.

Impact de l'Open-Shop et du Job-shop sur le C_{max} dans le cas général (toutes les expériences confondues) :

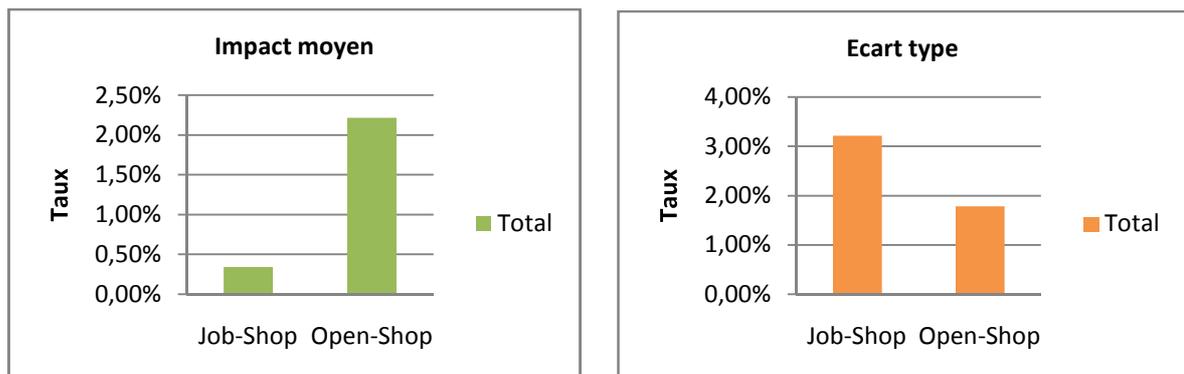


Figure VI.15 : Impact des expériences sur le C_{max} .

Conclusion :

Lors de l'analyse des taux d'amélioration des temps de séjours d'une commande dans la cellule flexible d'assemblage, nous avons constaté que l'amélioration du temps de séjour demeure toujours (temps d'exécution d'une commande) peu significative (inférieure à 5%) et vérifie toujours la conclusion énoncée par Finke et Dupont. La moyenne des taux de réduction ne dépassent pas 2.22% avec un écart type de l'ordre de 1.80%, obtenus par l'ordonnancement Open-Shop. La moyenne des taux d'amélioration dans le cas du Job-Shop est très faible et souvent négative.

On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible, ce qui signifie robustesse et stabilité des résultats, ainsi on peut conclure qu'indépendamment du nombre de stations on retrouve toujours le même comportement ainsi que les mêmes conclusions.

V.2. L'impact des règles de priorité au niveau des stations dans le cas Open Shop :

A partir des constats de la section précédente ; aucune amélioration quelle que soit l'ordonnement retenu, on se propose d'étudier le comportement du système géré par des plans non linéaires (Open Shop) combinés à un ordonnancement par règle de priorité auprès de chaque station. Les règles de priorité retenues sont :

- **FIFO** (Premier arrivé premier servi) ;
- **LPT** (la pièce qui a la durée opératoire la plus grande est prioritaire) ;
- **SPT** (la pièce qui a la durée opératoire la plus petite est prioritaire) ;
- **LRWK** (la pièce qui a le cumul de durées opératoires restantes le plus faible est prioritaire) ;
- **MRWK** (la pièce qui a le cumul de durées opératoires restantes le plus élevé est prioritaire).

Pour rappel, le nombre d'expériences menées dans cette partie est de 1620 (Résultats), chaque résultat est issu d'une simulation de 50 instances exécutées par le logiciel ARENA. Les paramètres de variation sont les suivant :

- Nombre de palettes : 10%, 15%, 20% et 25% de la taille de la commande ;
- Nombre de stations : 4, 5, 6, 7, 8, 9 et 10 ;
- Capacité des files d'attente d'entrée : 2, 4 et 6 entités ;
- Distance entre les stations : 5, 7 et 10 (l'unité est la palette) ;
- Et dans le cas à 4 stations la taille de la commande : 60, 80 et 100 entités (pièces à assembler).

Une fois montré que l'impact de la taille de la commande sur les performances de la cellule est insignifiant, la taille de la commande sera 20 fois le nombre de stations i.e. à 6 stations la taille de la commande sera $20 \times 6 = 120$.

On va ainsi analyser la performance globale (*C_{max}*) de la cellule induite par les différentes règles de priorité. Pour ce faire nous allons interpréter les temps de séjour dans la cellule d'un ensemble de pièces correspondant à une commande.

Analyse des temps d'assemblage d'une commande de pièces, minimisation du *C_{max}* :

L'analyse des résultats s'est faite par la lecture des graphes en considérant les paramètres de variation arrêtés précédemment.

Analyse préliminaire : Influence de la taille de la commande

Pour cela on a considéré la cellule flexible à 4 stations et nous avons considéré des commande de taille 60, 80, et 100 pièces, on a relevé les résultats suivants :

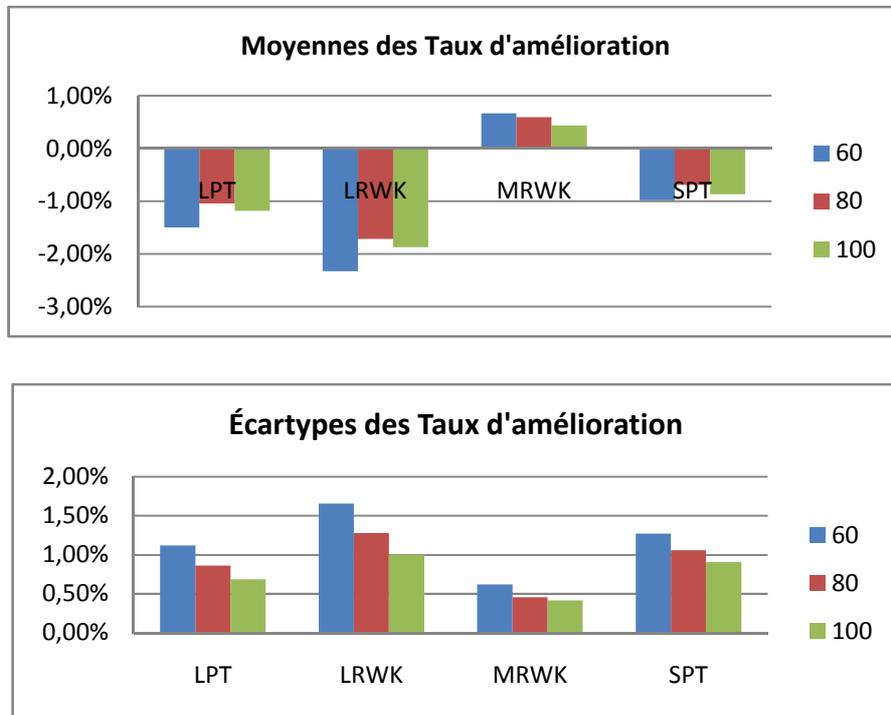


Figure VI.16: Taux d'amélioration du C_{max} selon la taille de la commande

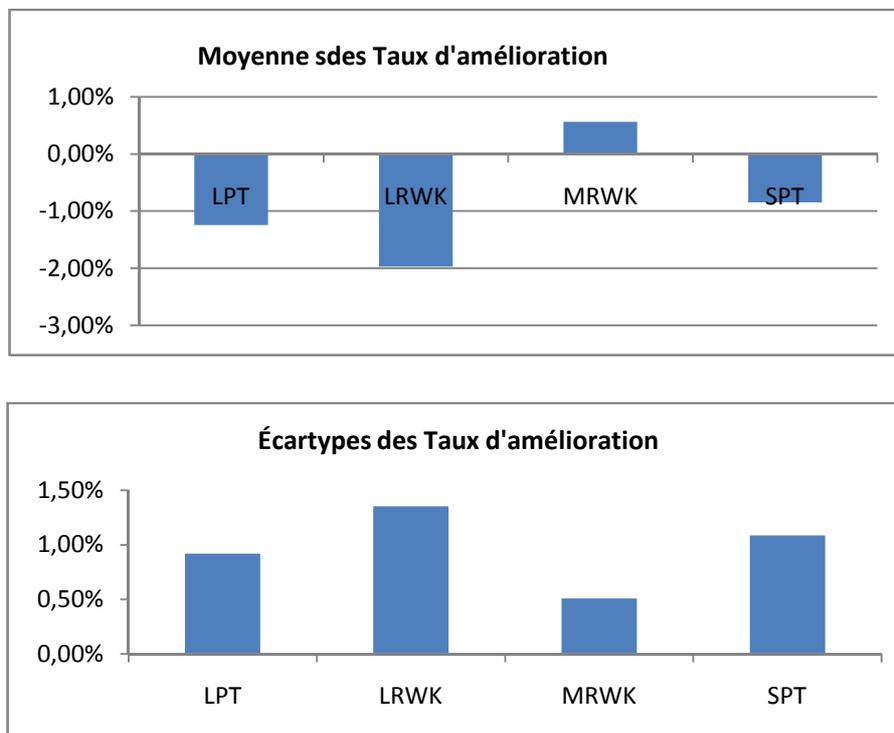


Figure VI.17 : Taux d'amélioration du C_{max} toutes tailles confondues.

D'après les résultats de la simulation (voir les graphes ci-dessus) l'impact de la taille de la commande sur la performance de la cellule d'assemblage est *peu significatif* (la dispersion est inférieure à 2%), et ce pour les quatre règles de pilotage. Dans les trois cas de figure (60, 80 et 100), la règle MRWK est la seule règle qui présente une amélioration, les autres induisent toujours une dégradation.

Du fait de l'impact faible de la taille de commande sur le C_{max} , elle ne sera plus considérée comme paramètre de variation mais calculée de la manière suivante :
 taille de la commande = $20 \times \text{nombre_stations}$.

On va analyser et comparer simultanément et l'impact des différents paramètres sur les performances enregistrées par chaque règle, et l'impact de chaque règle sur la performance de la cellule flexible.

Remarque : L'éclatement de l'analyse suivant les niveaux de chaque paramètre de variation sera effectué seulement pour le **cas 1** (4 stations), par la suite il est inutile d'aller vers plus de détails, et cela s'explique par la faiblesse de la dispersion autour des résultats moyens, d'où le peu de signification de l'impact des différents paramètres sur les performances de chaque règle.

Cas 1 : 4 stations

Impact de la capacité des files d'attente sur les performances des règles de priorité

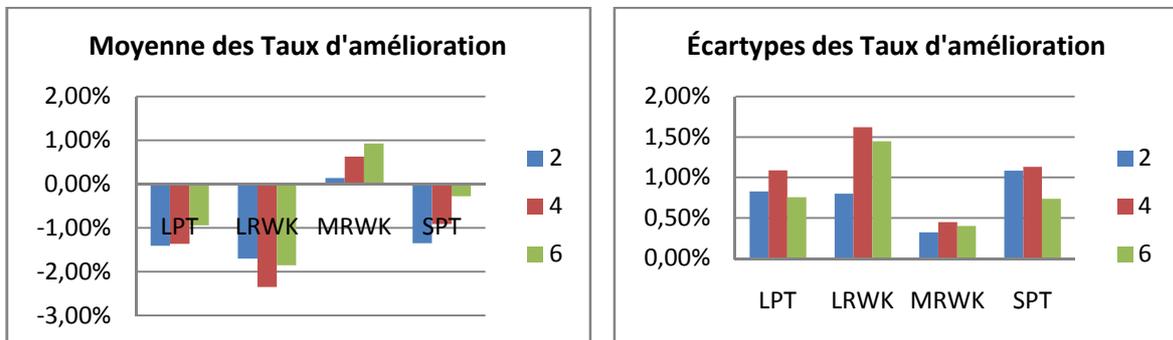


Figure VI.18 : Taux d'amélioration du C_{max} selon la capacité des files d'attente.

Nous remarquons que lorsque la file d'attente est égale à 6 on a les meilleures performances ou les plus faibles dégradations, cependant, l'ordre de grandeur de ces améliorations reste peu signifiant (inferieur à 2%). Nous déduisons de ce qui précède que pour chaque règle de priorité l'impact des capacités des files d'attente sur le C_{max} est quasiment inexistant.

Impact de la distance entre stations sur les performances des règles de priorité

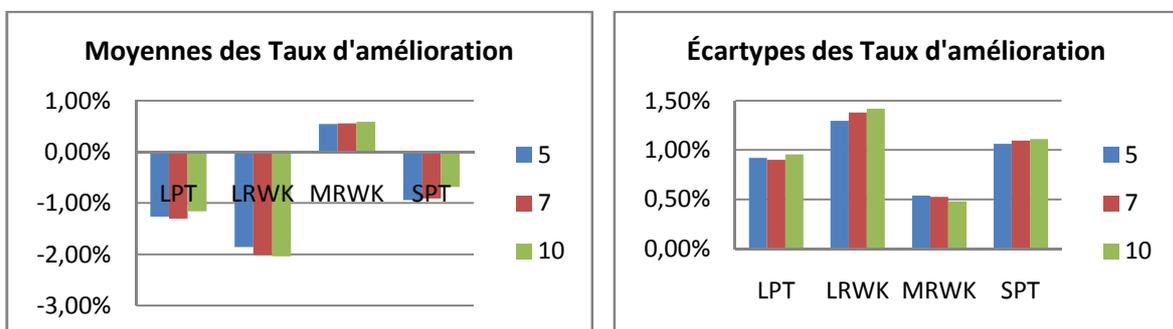


Figure VI.19 : Taux d'amélioration du C_{max} selon la distance entre stations.

Nous remarquons que l'ordre de grandeur des Moyennes des Taux d'amélioration est pratiquement le même pour les trois niveaux (5, 7 et 10). Nous déduisons de ce qui précède que pour chaque règle de priorité l'impact des distances entre stations sur le Cmax est quasiment inexistant.

Impact du nombre de palettes sur les performances des règles de priorité

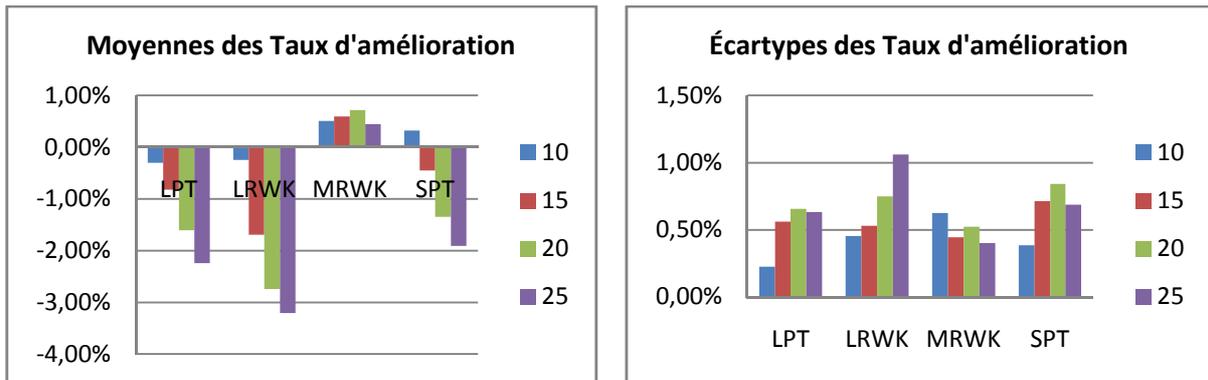


Figure VI.20 : Taux d'amélioration du Cmax selon le nombre de palettes.

Nous remarquons que pour les règles **LPT**, **LRWK** et **SPT**, la dégradation augmente avec le nombre de palettes. Pour la règle **MRWK** le nombre de palettes idéal est celui qui est entre 15 et 25% de la taille de la commande. Cependant l'ordre de grandeur de ces améliorations reste peu significatif (inferieur à 2%). Nous déduisons de ce qui précède que pour chaque règle de priorité l'impact du nombre de palettes sur le Cmax est quasiment inexistant.

Récapitulatif

Nous nous intéresserons à ce stade à l'impact des règles de priorité indépendamment des paramètres (files d'attente, distances,Etc.) puisque leur influence est insignifiante.

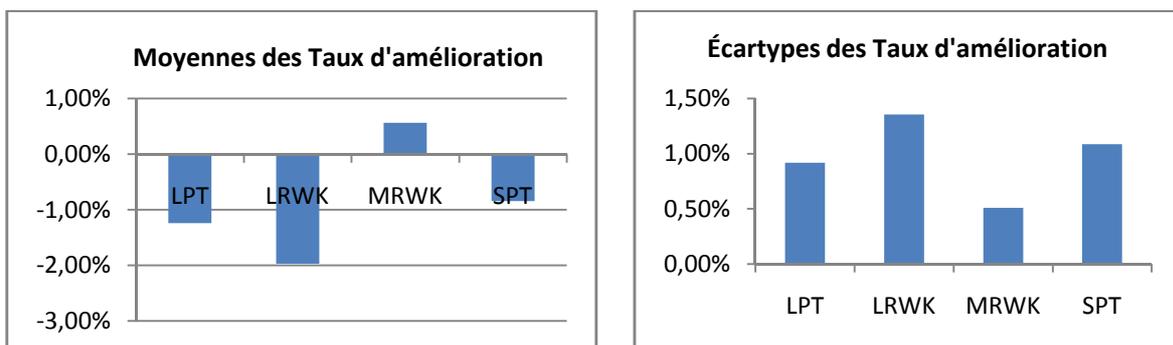


Figure VI.21 : Impact des différentes règles sur le Cmax.

Les trois règles **LPT**, **LRWK** et **SPT** présentent une dégradation du taux d'amélioration, seulement la règle **MRWK** présente une amélioration, cette dernière demeure peu significative (moyenne inferieure à 1%). On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible ce qui confirme à nouveau la stabilité des conclusions indépendamment des paramètres.

Cas 2 : 5 stations

A partir de ce cas seulement les résultats récapitulatifs qui tiennent compte de toutes les expériences seront présentés.

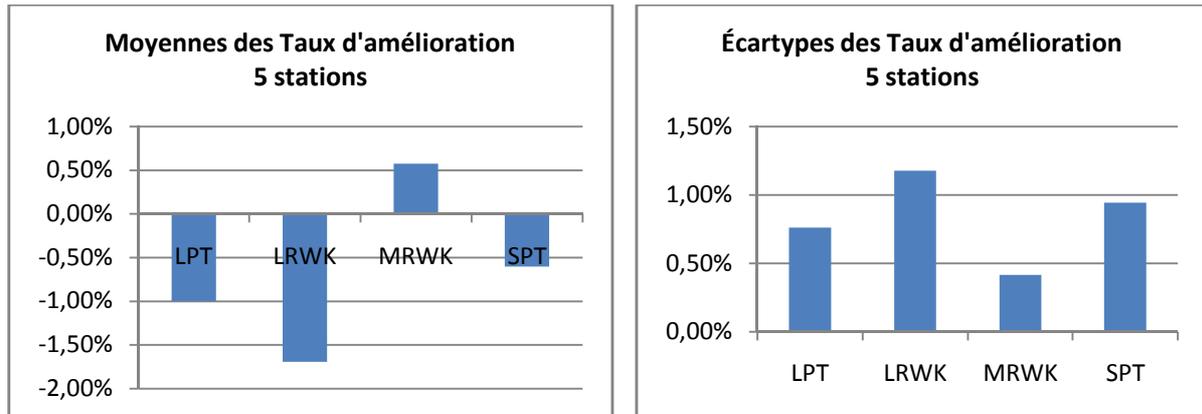


Figure VI.22 : Impact des différentes règles sur le Cmax.

A l'instar du cas précédent, l'expérience avec la configuration à 5 stations a abouti aux mêmes conclusions. Seulement la règle MRWK présente une amélioration, cette dernière demeure insignifiante (moyenne inférieure à 1%), les trois règles LPT, LRWK et SPT présentent une dégradation du taux d'amélioration et c'est à la règle LRWK que revient la plus mauvaise performance. On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible ce qui confirme à nouveau la stabilité des conclusions indépendamment des paramètres.

Cas 3 : 6 stations

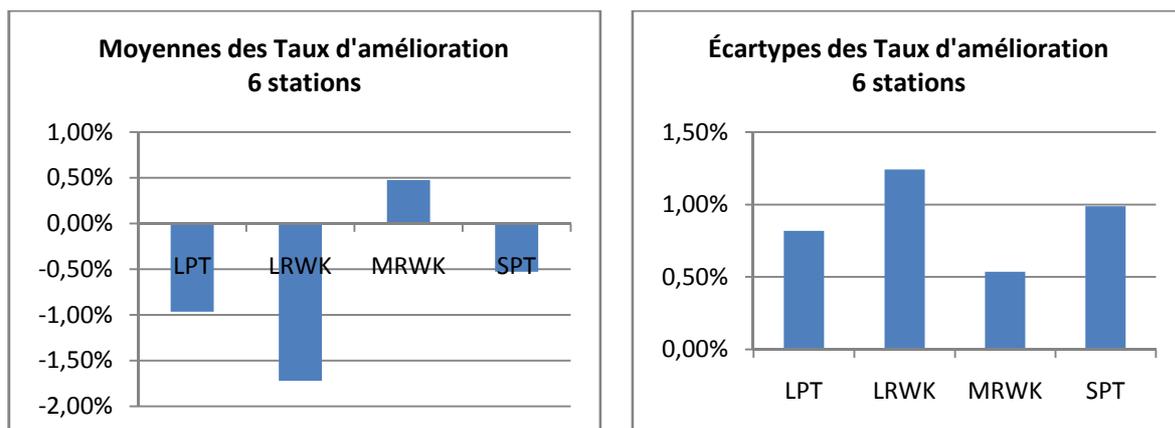


Figure VI.23 : Impact des différentes règles sur le Cmax.

Cas 4 : 7 stations

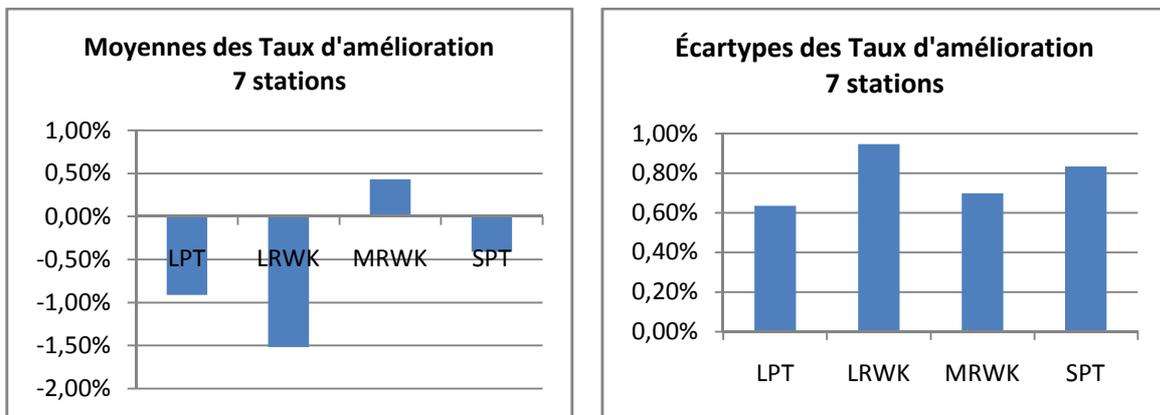


Figure VI.24 : Impact des différentes règles sur le Cmax.

Cas 5 : 8 stations

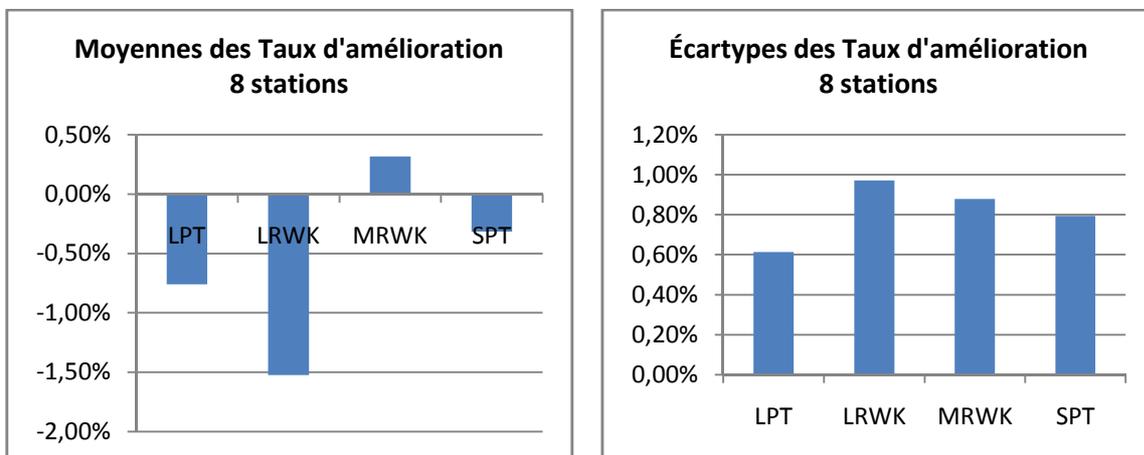


Figure VI.25 : Impact des différentes règles sur le Cmax.

Cas 6 : 9 stations

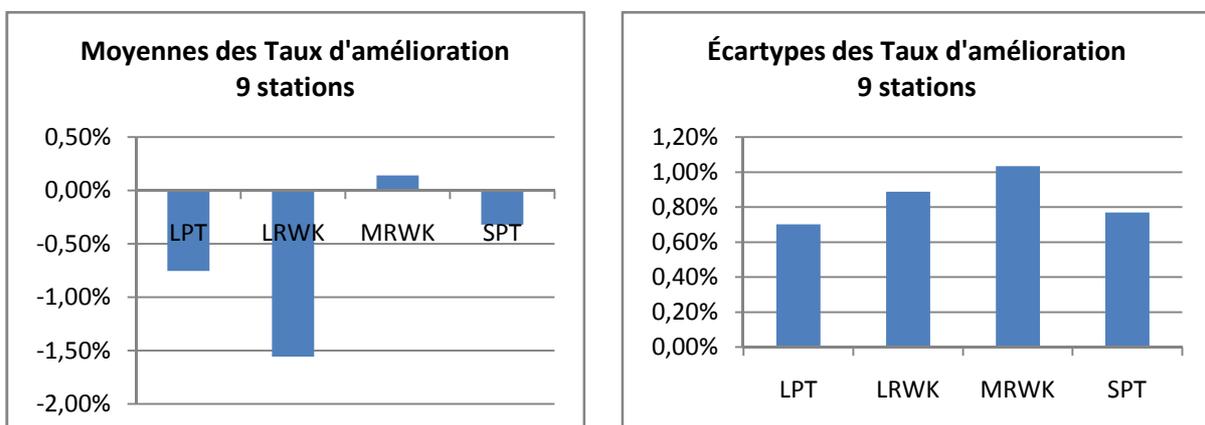


Figure VI.26 : Impact des différentes règles sur le Cmax.

Cas 7 : 10 stations

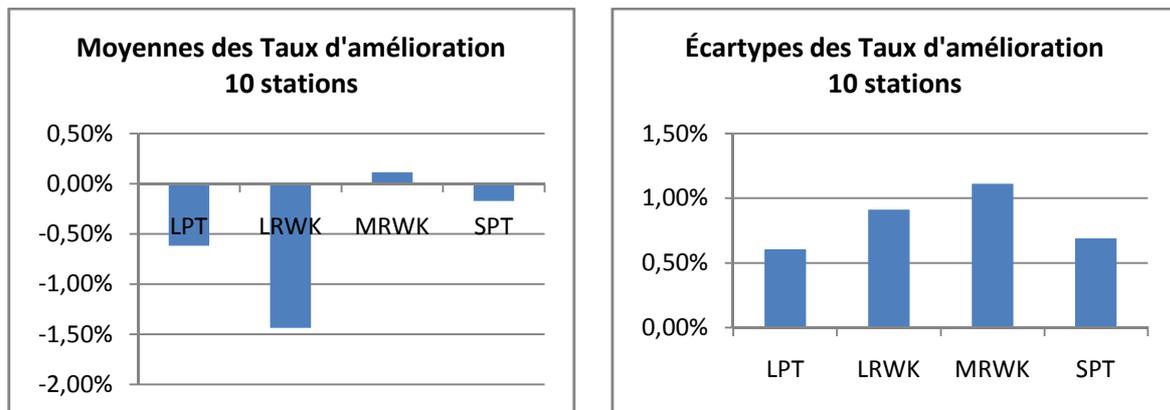


Figure VI.27 : Impact des différentes règles sur le Cmax.

Récapitulatif : A ce niveau nous allons synthétiser les résultats en considérant toutes les configurations possibles (4 stations,..., 10 stations).

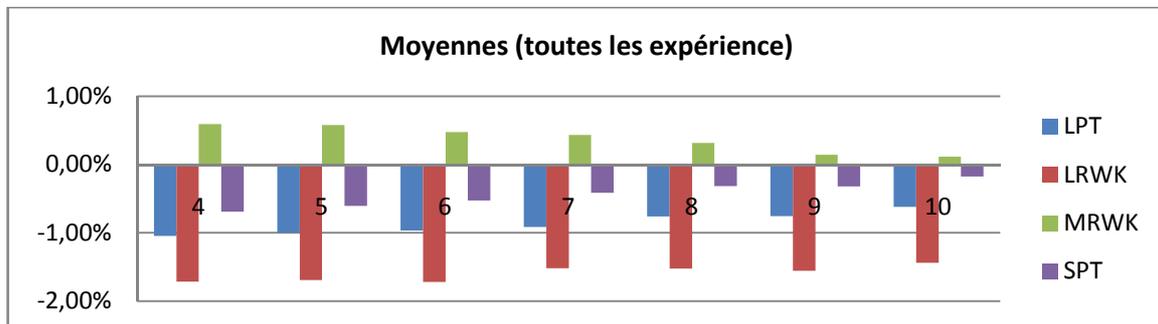


Figure VI.28 : Moyennes des taux d'amélioration.

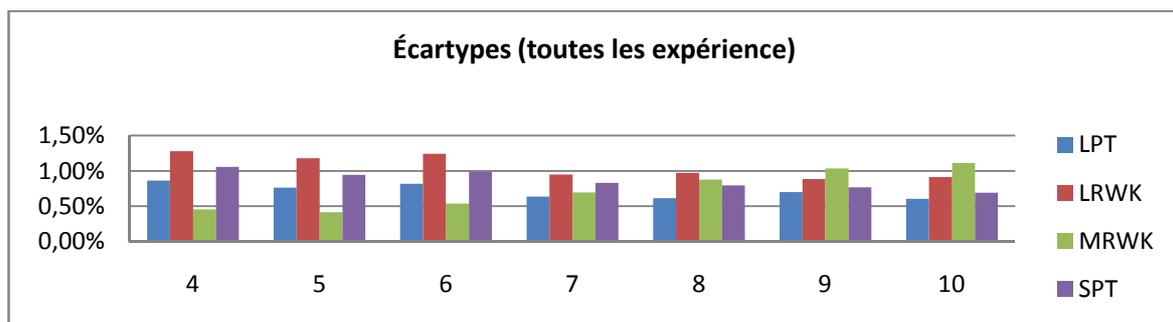


Figure VI.29 : Ecart type des taux d'amélioration.

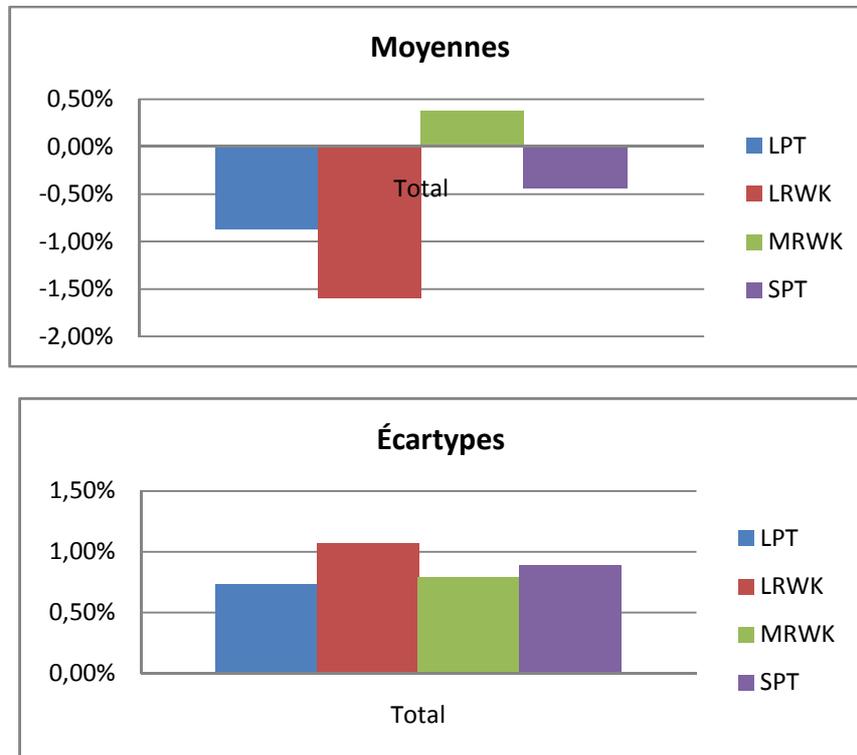


Figure VI.30 : Impact des règles de priorité (toutes les expériences).

Conclusion :

Ces quatre dernières figures consolident les conclusions obtenues dans les 7 derniers cas, ce qui nous mène à la conclusion finale :

Lors de l'analyse des taux d'amélioration des temps de séjours d'une commande dans la cellule, nous avons constaté que l'amélioration du temps de séjour demeure toujours peu signifiante (inférieur à 2%). Le nombre de dégradation est supérieur à celui des améliorations obtenues seulement dans le cas de la règle MRWK.

On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible (ce qui signifie robustesse et stabilité des résultats), ainsi on peut conclure qu'indépendamment du nombre de stations on retrouve toujours le même comportement ainsi que les mêmes conclusions.

V.3. L'approche collaborative :

Cette partie est réservée à l'analyse de l'impact de l'approche collaborative sur les performances de la cellule flexible d'assemblage à 4 stations, règle de priorité FIFO.

Impact de la capacité des files d'attente sur les performances de l'approche :

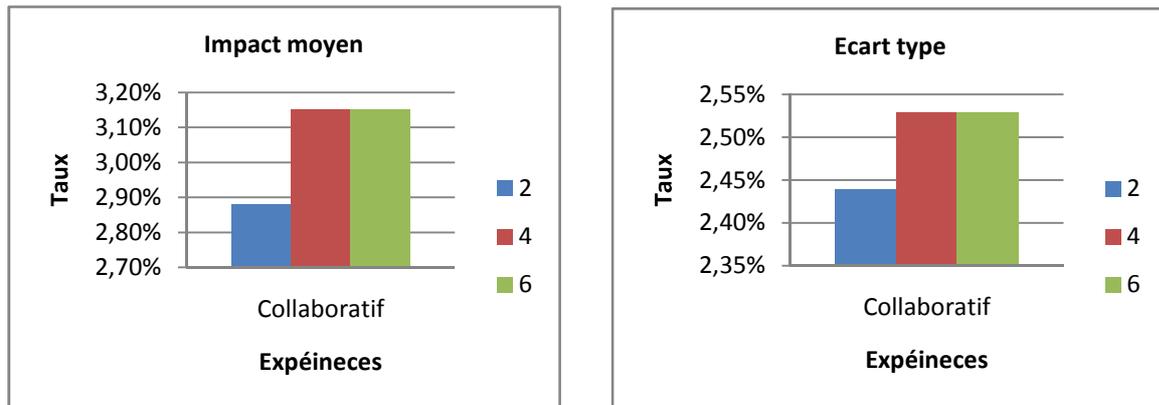


Figure VI.31 : Impact de la capacité des files d'attente.

D'après ces résultats (graphe ci-dessus), la moyenne des taux d'amélioration ne dépasse pas 3.15% avec un écart type de 2.53% obtenu dans le cas où les files d'attente sont à 4 et 6 entités.

Nous remarquons une amélioration des performances (**Cmax**) proportionnellement aux capacités des files d'attente, mais cet impact reste très faible.

Impact de la distance entre stations:

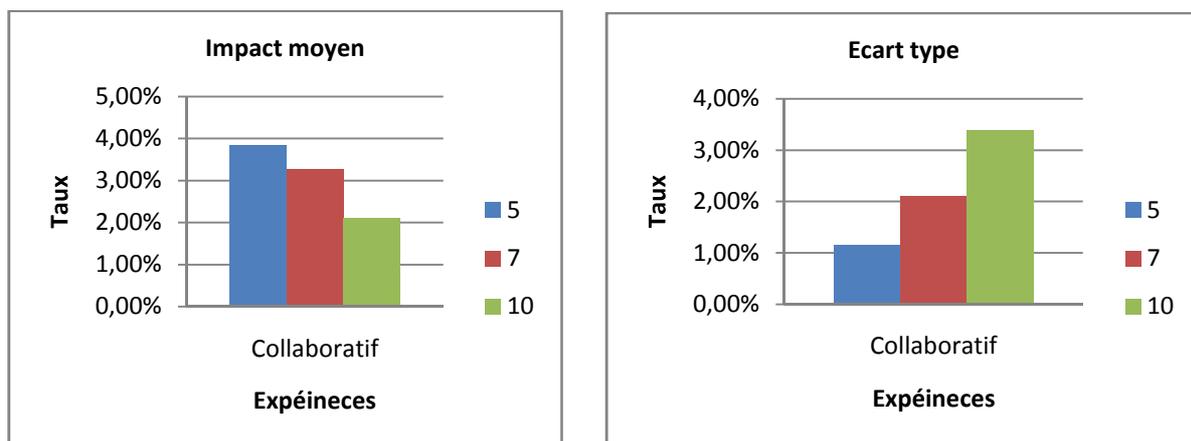


Figure VI.32 : Impact de la distance entre stations.

Nous remarquons que la moyenne des taux d'amélioration ne dépasse pas **3.83%** avec un écart type de **1.15%** obtenu dans le cas de l'approche collaborative de distance entre stations égal à 5 palettes. L'impact est toujours insignifiant.

Impact du nombre de palettes:

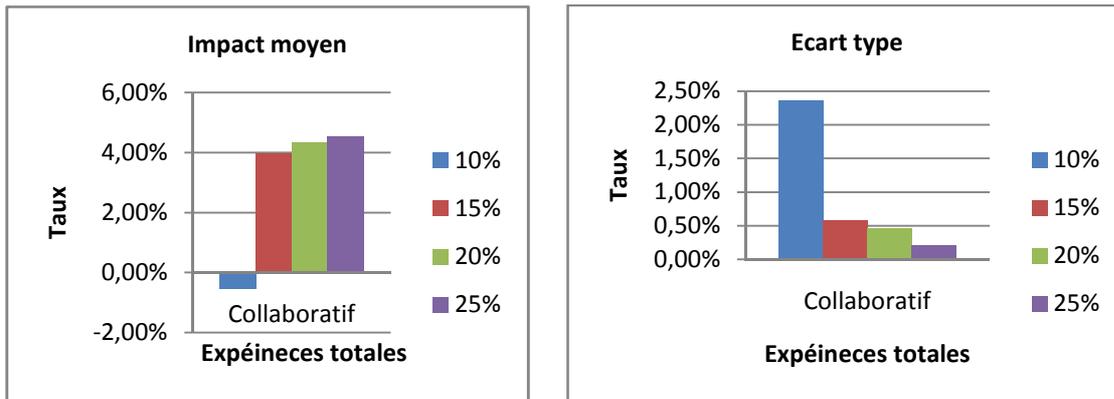


Figure VI.33 : Impact du nombre de palettes.

D'après les graphes ci-dessus, la moyenne des taux d'amélioration ne dépasse pas **4.51%** avec un écart type de **0.20%** obtenu dans le cas de l'approche collaborative avec un nombre de palette de 25% ($0.25 \times 80 = 20$ palettes).

Impact de l'approche collaborative sur le *Cmax* :



Figure VI.34 : Impact de l'approche collaborative sur le *Cmax*.

Conclusion :

Le pilotage de la cellule par l'approche collaborative améliore le C_{max} ($3.06\% \pm 2.43\%$) mais son impact est également peu significatif comme le pilotage Open-Shop.

On remarque par ailleurs que la dispersion des résultats autour des moyennes est très faible, ce qui confirme à nouveau la stabilité des conclusions indépendamment des paramètres de variation.

VI. Conclusion

A la lumière des conclusions tirées ci-dessus, on peut déduire qu'aucune règle de priorité ni organisation d'atelier n'améliore d'une manière significative le temps de séjour d'une commande (C_{max}) dans le système étudié.

Conclusion générale

Dans le cadre de ce travail, nous avons traité des problèmes d'ordonnancement d'un atelier d'assemblage dans les environnements Open-shop et Job-Shop (les organisations d'ateliers les plus adaptés aux SFP). Le système étudié est une cellule flexible d'assemblage dont les stations (variant de 4 à 10 stations) sont conçues autour d'un convoyeur en boucle avec accumulation. Nous avons étudié d'abord le comportement d'un tel système (**C_{max}**) géré par des plans non linéaires prédéterminés puis non prédéterminés (dans le cas à 4 stations, nous avons même considéré une approche collaborative), la règle de priorité au niveau des files d'attente est FIFO. Nous avons étudié l'apport de tels ordonnancements par rapport aux performances enregistrées pour un ordonnancement Flow-Shop d'un même système. Par la suite, nous avons étudié l'impact des règles de priorité sous un ordonnancement Open-Shop. Par l'application de ces règles de priorité, l'objectif était la réduction des temps de séjour d'une commande de pièces (**C_{max}**) par rapport aux temps obtenus sous un pilotage FIFO.

Pour ce faire, nous avons conçu un modèle de simulation *flexible* et défini une planification d'expériences basée sur le nombre de stations, la capacité des files d'attente, la longueur du convoyeur, le nombre de palettes, les règles de priorité ainsi que le type d'ordonnancement. En effet, le modèle nous permet grâce à une interface graphique (tableau de bord) tout à fait au début, de définir le niveau désiré de n'importe quel paramètre défini précédemment (ainsi que d'autres options). Et grâce à une autre interface à la fin de la simulation, nous visionnons la performance enregistrée avec un rappel des niveaux des différents paramètres choisis (pour le cas simulé).

En tout, nous avons effectué **2700** expériences nettes (**135000** instances). L'analyse des résultats de l'expérimentation a bien montré que les taux de réduction des temps de séjours d'une commande (**C_{max}**) en considérant le type d'ordonnancement et par la suite le seul fait de l'application des règles de priorités étaient peu significatifs (< 5%) :

Considération du type d'ordonnancement :

La moyenne des taux de réduction (amélioration) ne dépassent pas **2.22%** avec un écart type de l'ordre de **1.80%**, obtenus par l'ordonnancement Open-Shop. La moyenne des taux d'amélioration dans le cas du Job-Shop est très faible et souvent négative (**0.34% ± 3.22%**).

Le pilotage de la cellule par l'approche collaborative améliore le **C_{max}** (**3.06% ± 2.43%**) mais son impact est également peu significatif comme le pilotage Open-Shop.

Considération des règles de priorité :

Nous avons constaté que l'amélioration du temps de séjour reste toujours peu significative (inférieur à **2%**). Le nombre de dégradation est supérieur à celui des améliorations obtenues seulement dans le cas de la règle MRWK.

Conclusion générale

On remarque aussi que la dispersion des résultats autour des moyennes est très faible (robustesse et stabilité des résultats). Ainsi, on peut conclure qu'indépendamment du nombre de stations on retrouve toujours le même comportement ainsi que les mêmes conclusions.

Cette étude nous a initiés à la conduite de projets de simulation des systèmes de production, voire maîtriser le processus de simulation. En effet, la simulation est la méthode la plus adaptée à l'étude du comportement de ce type de systèmes, qui permet de considérer toutes les contraintes, sans imposition de restrictions, tel que c'est le cas lors de l'utilisation des méthodes analytiques.

Au vu des résultats obtenus et analysés, nous pensons qu'un ordonnancement coopératif sera plus judicieux et plus efficace, et ceci par la prise en compte de l'état en cours de chaque ressource (de transformation, de stockage et de manutention). L'idéal serait, qu'à chaque point d'intersection (entrée de la cellule-convoyeur, convoyeur-entrée d'une station), on puisse calculer toutes les trajectoires possibles (tout en considérant les temps d'attente probables dans les files d'attente d'entrée et les règles de priorité appliquées, les distances entre les points d'entrée des stations, les taux d'exécution des tâches en cours) et d'en choisir la meilleure. Ceci implique l'élaboration d'un protocole de communication performant entre les points d'accès aux stations.

L'objectif sera de réaliser un ordonnancement de la cellule (les stations allant de 4 à 10) d'une performance globale satisfaisante tout en respectant l'autonomie décisionnelle de chaque ressource de la cellule. Ceci fera l'objet de travaux futurs.

Liste des annexes

Annexe I : Définitions, notations des problèmes d'ordonnancement.

Annexe II : Illustration d'un exemple Arena, l'approche par processus.

Annexe I

1. Définitions :

Conduite : Elle se charge de la traduction de l'ordre de fabrication en une séquence d'instructions exécutables par une ressource.

Contraintes cumulatives : Elles sont liées à l'utilisation simultanée d'une même ressource par plusieurs tâches et indiquent que leur capacité est limitée.

Contraintes disjonctives : Spécifient que le traitement d'un ensemble de tâches sur la ressource disjonctive ne peut se faire simultanément au même instant.

Contraintes physiques : Spécifiant les caractéristiques limitant le fonctionnement de l'atelier, dues à l'organisation physique des moyens de production.

Contraintes temporelles : Elles regroupent les contraintes de précédences et de localisation temporelle.

- Les contraintes de précédences entre les tâches prennent en compte la succession des opérations de la gamme opératoire. Ces contraintes, indiquant que la date de début d'une tâche j doit commencer après la date de fin de réalisation de la tâche i ;
- Les contraintes de localisation temporelle définissent l'intervalle de temps sur lequel la tâche doit être traitée. Cet intervalle est limité par les valeurs des dates de début au plutôt et de fin au plus tard.

Gamme : Un document décrivant en détail la séquence d'opérations de fabrication, d'assemblage, d'inspection ou de transport nécessaires à la fabrication d'un composant produit fini.

Gammes libres : L'exécution des opérations est indépendante de l'ordre, il n'existe aucune contrainte de succession. Les gammes libres caractérisent les problèmes d'ordonnancement du type Open Shop que nous définiront plus loin.

Gammes linéaires : L'ordre d'exécution des opérations est entièrement imposé et prédéterminé. Ce genre de gammes est présent dans le cas du job shop (chaque tâche à sa propre route à suivre autrement dit toutes les tâches ont des gammes prédéterminées mais non identiques) et celui du Flow Shop ou toutes les tâches ont le même chemin à suivre (la même route ou la même gamme).

Gammes mixte ou semi-linéaires : L'ordre d'exécution des opérations est partiellement déterminé, on trouve des tâches qui ont des gammes prédéterminées (on sait a priori la route à suivre par ces tâches), et d'autre non (on ne sait pas, a priori, la route à suivre par ces tâches)

Gammes techniques : Elles sont dues essentiellement au choix prédéterminé de certaines tâches à passer sur des machines spécifique.

Pareto-dominance : Une solution est Pareto-dominante si aucune autre solution n'améliore la satisfaction d'un critère sans dégrader la satisfaction d'un autre.

Productique : C'est l'ensemble des sciences et techniques ayant pour but d'obtenir le maximum d'efficacité des systèmes de production. Elle ne se limite donc pas à une technique ou à une méthodologie, c'est une approche globale.

Ressource (1) : Tout moyen à disposition d'une entreprise pour la production et la livraison d'un produit ou d'un service.

Ressource (2) : est un objet (par exemple une machine, un ordinateur, un lingot de métal) ou un service (par exemple le travail d'un ouvrier, une étude soustraitée), porteur de valeur, consommé ou consommable par l'entreprise dans le cadre de l'un de ses processus. La consommation de la ressource est mesurable physiquement (heurs, kilos) ou monétairement. La mesure monétaire d'une consommation de ressource est un coût.

Ressources disjonctives : (non partageables) principalement dans le cas de ressources renouvelables, sont des ressources qui ne peuvent exécuter qu'une tâche à la fois (machine outil, robot manipulateur, etc.). C'est le cas des ressources de notre problème.

Ressources consommables : Une ressource est consommable si elle devient non disponible après avoir été utilisée par une ou plusieurs tâches (matière première, budget, etc.), la consommation globale au cours du temps est limitée.

Ressources cumulatives (partageables) : sont des ressources qui peuvent être utilisées par plusieurs tâches simultanément (équipe d'ouvriers, poste de travail).

Ressources doublements contraintes : lorsque son utilisation instantanée et sa consommation globale sont toutes les deux limitées (sources d'énergie, financement, etc.).

Ressources renouvelables : Une ressource est renouvelable si, après avoir été utilisée par une ou plusieurs tâches, elle est à nouveau disponible en même quantité (les hommes, les machines, l'espace, l'équipement en général, etc.). La quantité de ressource utilisable à chaque instant est limitée.

La robustesse d'un système de production se définit par son aptitude à produire conformément aux résultats attendus. Cela suppose la garantie de l'obtention des performances souhaitées en présence d'incertitudes dans le système.

2. La notation en $\alpha/\beta/\gamma$ des problèmes d'ordonnement : (tableau AI)

α	Système	α_1	\emptyset	une seule machine		
			P	machines identiques en parallèle		
			U	machines à vitesse proportionnelle en parallèle		
			R	machines non reliées en parallèle		
			O	open shop		
			F	flow shop		
			J	job shop		
		α_2	\emptyset	le nombre de machines est fixé à une valeur donnée précise		
			1,2...	le nombre de machines est fixé dans la définition de l'énoncé		
			m	le nombre de machines varie avec chaque instance considérée		
β	Contraintes	β_1	\emptyset	les opérations ne peuvent pas être interrompues		
			<i>pmtn</i>	les opérations peuvent être interrompues		
		β_2	\emptyset	il n'y a pas de contraintes de précédence		
			<i>tree</i>	il existe des contraintes de précédence additionnelles sous forme d'arborescence ou d'anti-arborescence		
			<i>prec</i>	il existe des contraintes de précédence additionnelles sous forme d'un graphe orienté sans circuit		
		β_3	\emptyset	$\forall i r_i = 0$		
			r_j	les dates de disponibilité sont différentes selon les travaux		
		β_4	\emptyset	$\forall i p_i \in \mathbb{N}^*$		
			$p_i=1$	chaque opération a une durée unitaire		
			$(p_j=p) \& (\beta_3= r_j)$	La durée des opérations est égale à $p \neq 1$ et au moins un des r_j n'est pas divisible par p		
		β_5	\emptyset	les temps de réglage sont constants et inclus dans la durée des opérations		
			$Snsd^1$	des temps de réglage sont affectés à chaque opération, ils sont indépendants de la séquence mais non inclus dans la durée opératoire car ils peuvent être exécutés avant que l'opération n'arrive dans la file d'attente de la machine		
			Ssd^1	les temps de réglage ne sont pas inclus dans la durée des opérations et sont dépendants de la séquence des opérations sur la ressource		
		β_6	\emptyset	les dates de fin souhaitées si elles existent sont des nombres entiers arbitraires non négatifs		
			$d_j=d$	toutes les dates de fin souhaitée sont identiques (« common due date »)		
			Δ_j	les dates de fin souhaitée sont remplacées par des dates de fin impérative		
		γ	Critères	γ_1	C	Completion time
					L	Le retard « Lateness »
					T	Le retard vrai « Tardiness »
					E	L'avance vraie « earliness »
G	une fonction générale non décroissante du « completion time »					
NT	le nombre de travaux en retard					
γ_2	K_{max}			Minimisation de la plus grande valeur d'une fonction donnée de la date de fin pour tous les travaux ($K \equiv C, L, T, E, G$)		
	\bar{K}			Minimisation de la valeur moyenne d'une fonction donnée de la date de fin pour tous les travaux ($K \equiv C, L, T, E$)		
	\bar{K}_w			Minimisation de la somme pondérée de la fonction considérée de la date de fin pour tous les travaux ($K \equiv C, L, T, E$)		

Annexe II

L'approche Processus (Process Interaction) :

Elle consiste à décrire le fonctionnement d'un système comme une interaction de plusieurs processus, à travers des primitives, des blocs ou des fonctions, conçus au préalable. Souvent, on considère dans ce type de logique (modélisation) la notion de file d'attente qui introduit certains problèmes dans la modélisation des systèmes sans stocks.

Le logiciel SIMAN (Pegden, 1990), privilégie l'approche processus. Un processus est une séquence de transformations que subit une entité pendant son séjour dans le système. Une entité est un objet dont l'état est modifié par le processus qu'il traverse. Les changements d'état nécessitent souvent l'utilisation de ressources. Un modèle est constitué par un ensemble de processus qui interagissent entre eux.

La construction du modèle logique orienté processus peut dès lors s'articuler en plusieurs étapes logiquement successives :

— **Identifier les entités :**

Les entités physiques sont en général facilement repérables, pièces dans un atelier, clients dans un magasin, automobiles dans une station service...

— **Identifier les processus :**

Processus d'entrée, de transformation, de destruction...

— **Identifier les ressources :**

Tous les objets dont les entités ont besoin au cours de leur transformation : ressources de stockage dans lesquelles sont mises en attente d'un événement, et les ressources de transformation à proprement parler qui dépendent du processus étudié.

— **Identifier les variables d'état :**

Elles caractérisent le système dans son évolution au cours du temps. Elles sont de deux types :

Les attributs : ce sont des variables spécifiques à une entité donnée et qu'elle comporte en elle pendant sa durée de vie dans le système (type d'entité, gamme opératoire...). Ces variables sont locales.

Les variables globales : sont celles qui, au contraire, sont connues de tous les processus.

— **Décrire les processus :**

A ce stade, il ne reste plus qu'à assembler tous les éléments précédemment définis dans un modèle logique approprié composé de processus parcourus par des entités utilisant les ressources et se synchronisant entre eux.

L'approche par processus est l'une des quatre approches de simulation suivantes :

— Approche orientée événements (event scheduling) ;

Annexe II

- Approche orientée activités (activity scanning) ;
- Approche orientée processus (process interaction) ;
- Approche orientée objets.

Nous avons opté pour la troisième approche car c'est elle qu'utilise le logiciel de simulation ARENA 12 et elle est la plus utilisée par les langages de simulation.

Bibliographie

1. Ouvrages

- (Bellman, 1957) R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.
- (Botta, 1999) G. Botta, *Gestion de Production*. INSA, Département de Génie Productique. 1998-1999.
- (Carlier et Chrétienne, 1988) J. Carlier et P. Chrétienne, *Problèmes d'Ordonnancement : Modélisation, Complexité et Algorithmes*. Paris, Masson, 326p, 1988.
- (Courtois et al, 2002) A.Courtois, C.Martin-Bonnefous, M. Pillet « *Gestion de la production* », 3^{ème} Edition, Collection EO SUP, Editions d'Organisation, 2002.
- (Esquirol et Lopez, 1999) P. Esquirol et P. Lopez, *L'Ordonnancement*, Economica, 1999.
- (Giard, 2003) Giard V, "*Gestion de la production et des flux*", 3^{ème} édition, Collection Gestion, Série : Production et techniques quantitatives appliquées à la gestion, Economica, 2003.
- (GOTHA, 1993) GOTHA, *Les problèmes d'ordonnancement*, volume 27, n°1, chapitre 4, pages 77_150. R.A.I.R.O, Recherche Opérationnelle/Operational Research, 1993.
- (Habchi, 2001) G. Habchi *Conceptualisation et modélisation pour la simulation des systèmes de production ; HDR ; Université de Savoie, France, 2001.*
- (Kelton, 2004) Kelton W.D., 2003, "*Simulation with Arena*", 3^{ème} Edition, International Edition, New York.
- (Kieffer et Pujo, 2002) Kieffer J., Pujo P, « *Fondements du pilotage des systèmes de production* », Edition Lavoisier, Paris, 2002.
- (Parrish, 1993) D.Parrish, « *Flexible Manufacturing* », Edition Butterworth-heinemann, 1993.
- (Le Moigne, 1990) J.L. Le Moigne, « *La théorie du système général : théorie de la modélisation* », 2^{ème} édition, Paris, France, 1990.
- (Lenstra et Rinnoy Kan, 1978) J. Lenstra and A. Rinnoy Kan, *Complexity of scheduling under precedence constraint*, Operations Research, 22-35, 1978.
- (Lopez et Roubllat, 2001) P. Lopez et F. Roubllat, *Ordonnancement de la production*, Paris Hermès, 2001.
- (Melnyk et Carter, 1987) S. A. Melnyk and Phillip L. Carter, *Production Activity Control*, The Business ONE Irwin/APICS Series in Production Management, Richard D. Irwin, Inc., 1987
- (Nand, 1991) Nand K.JHA, « *Handbook of Flexible Manufacturing Systems* », Edition Academic Press, INC, California, 1991.

Références bibliographiques

- (Pegden, 1990) C.D. Pegden, R.E. Shannon, R.P. Sadowski, « Introduction to simulation using SIMAN », McGraw Hill, New York, NY, 1990.
- (Pinedo, 1995) M. Pinedo, *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, 1995.
- (Pritsker, 1986) A.A.B. Pritsker, « Introduction to simulation and SLAM II », Halsted Press, New York, NY, 3rd edition, 1986.
- (Rayward- Smith et Rebaine, 1992) V.J. Rayward-Smith, D. Rebaine, *Open-shop scheduling with delays*, Theoret. Inform. Appl. 439-448, 1992.
- (Rodde, 1991) G. Rodde, *Les systèmes de production – modélisation et performance* ; Éditions Hermès, 1991.
- (Sahni et Cho, 1979) S. Sahni and Y. Cho; *Complexity of scheduling shops with no wait in process*; Math. Oper. Res., 4:448-457, 1979.
- (Zeigler, 1976) B. P. Zeigler; *Theory of Modeling and simulation*. New-York: John Wiley & Sons, 1976.

2. Revues

- (Bel, 1991) G. Bel, Rapport introductif du thème « *Pilotage et gestion d'atelier* », Colloque Actions de recherche en Robotique et Productique du MRT, Paris, pp. 551-562, 1991.
- (Bel et Dubois 1985) G. Bel, G. Dubois, *Modélisation et simulation des systèmes de production*, Revue APII n°19, 1985.
- (Billaut, 1999) J.C. Billaut, *Recherche Opérationnelle et aide à la décision pour les problèmes d'ordonnancement*, Laboratoire d'Informatique, Université François Rabelais, Tours, France, 1999.
- (Cathy et Liu, 1993) B.L. MacCathy and J. Liu, *Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling*, volume vol. 31, no. 1, pages 59-79, International Journal of Production Research, 1993.
- (Dell'Amico et Vaessens, 1995) M. Dell'Amico, R.J.M. Vaessens, *Flow-Shop and Open-Shop scheduling on two machines with transportation times and machine-independent processing times is NP-hard*, Materiali di discussione 141, Dipartimento di Economia Politica, Università di Modena, Italy, 1995.
- (Draghici et al, 1998) Draghici G., Brnzei N., Filipas I, « *La modélisation et la simulation en vue de la conduite des systèmes de production* », Les Cahiers des Enseignements Francophones en Roumanie, universitatea politehnica Din Timisoara, 1998.
- (Dupont et al, 1997) L. Dupont, Y. Crama, G. Finke, *Recherche Opérationnelle et gestion de la production*, Article préparé dans la revue : nouvelles de la science et des technologies. 1997.
- (Finke et Dupont, 1993) G.Finke, L. Dupont , *Combinatorics of Cyclic Assembly Systems*, IEEE Systems, Man and Cybernetics Conference Proceedings, Laboratoire ARTEMIS-IMAG,Grenoble, 1993.
- (Gonzales et Sahni, 1976) T. Gonzales and S. Sahni, *Open shop scheduling to minimize finish time*, Journal of the Association for Computing Machinery, 23(4):665-679, October 1976.

Références bibliographiques

- (Grabot, 2005) B. Grabot, *Ordonnancement d'ateliers manufacturiers*, Techniques de l'ingénieur, AG 3 015.
- (Jubin, 2003) Jubin M., 1994, « *Ateliers flexibles d'usinage* », Edition Technique de l'Ingénieur, B 7030.
- (Trentesaux et al, 1996) D. Trentesaux, C. Tahon, *HPACS, une approche égalitaire supervisée et interactive pour le pilotage adaptatif d'un système de production*, Journées Ordonnancement Coopératif, Bordeaux, 1996.

3. Mémoires

- (Bakalem, 1996) Bakalem M., « *Modélisation et simulation orientées objet des systèmes manufacturiers* », Thèse de Doctorat en Electronique-Electrotechnique-Automatique, Université de Savoie, France, 1996.
- (Behiri et latreche, 2009) W. Behiri, M. R. Latreche, *Contribution à la résolution d'un problème d'ordonnancement en temps réel sur deux ressources identiques en parallèle par l'approche collaborative*, Projet de Fin d'Étude en Génie Industriel, juin 2009.
- (Beslon, 1995) Beslon G, « *approche pour le pilotage réactif en atelier flexible* », Thèse de doctorat : Ingénierie Informatique, l'institut national des sciences appliquées de Lyon, France, 1995.
- (Caumon, 2006) Caumon A, 2006, « *le problème de jobshop avec contraintes: modélisation et optimisation* », Thèse de doctorat, Université Blaise Pascal, France.
- (Djellouli, 2007) S. Djellouli, *Approche par coopération inter machine pour un ordonnancement distribué*, Thèse de Magistère en Automatique, 2007.
- (Esswein, 2003) C. Esswein, *Un apport de flexibilité séquentielle pour l'ordonnancement robuste*, Thèse de Doctorat, Université François Rabelais, Tours, France, 2003.
- (Gazoby, 2003) S. H. Gazoby, *Planification des tâches dans un environnement d'ateliers à flux tirés*, Mémoire de Magister, EMP, Algérie, 2003.
- (Ghazi, 2007) Ghazi N, « *Problème d'ordonnancement Open Shop dans un environnement d'atelier d'assemblage cyclique* », mémoire de magister, Spécialité : automatique, département Génie électrique, Laboratoire de Commande des Processus, ENP, Alger, 2007.
- (Gueret, 1997) C. Gueret, *Problèmes d'ordonnancement sans contraintes de précédence*, Thèse de doctorat, Université de Technologie de Compiègne, France, 1997.
- (Hentous, 1999) H. Hentous, *Contribution au pilotage des systèmes de production de type Job Shop*, Thèse de doctorat INSA Lyon, 1999.
- (Letouzey, 2001) Agnès Letouzey; *Ordonnancement interactif basé sur des indicateurs : Applications à la gestion de commandes incertaines et à l'affectation des opérateurs*; Thèse de Doctorat, INP, Toulouse, France, 2001.
- (Nagi, 1991) G.Nagi, *Design and operation of hierarchical production manufacturing systems*, PhD, Thesis Report, University of Maryland, USA, 1991.

“Puisqu'on ne peut être universel en sachant tout ce qui se peut savoir sur tout, il faut savoir peu de tout, car il est bien plus beau de savoir quelque chose de tout que de savoir tout d'une chose. Cette universalité est la plus belle. Si on pouvait avoir les deux encore mieux, mais s'il faut choisir il faut choisir celle-là. Pensées, Et le monde le sait et le fait, car le monde est un bon juge souvent”

[Blaise Pascal (1623-1662)]

“En essayant continuellement, on finit par réussir, donc : plus on rate, plus a de chances que ça marche”

[Les Shadockç.]