

Contribution à l'optimisation de performances de systèmes de production flow-shop coordonnés par un contrôleur central

Propositions pour un ordonnancement efficace
de l'activité d'une ressource partagée

THÈSE

présentée et soutenue publiquement le 27 mai 2008

pour l'obtention du

Doctorat en Sciences de l'Université de Bretagne Occidentale

(Discipline : Informatique)

par

Loïc PLASSART

Composition du jury

<i>Président :</i>	Jacques TISSEAU	Professeur des Universités (ENIB – LISyC)
<i>Rapporteurs :</i>	Maryline CHETTO	Professeur des Universités (Univ. de Nantes – IRCCyN)
	Jean-Louis FERRIER	Professeur des Universités (Univ. d'Angers – LISA)
<i>Examineurs :</i>	Jean-Charles AKIF	Responsable d'Études (CNAM des Pays de la Loire)
	Philippe LE PARC	Professeur des Universités (UBO – LISyC)
	Frank SINGHOFF	Maître de Conférences (UBO – LISyC)
<i>Invité :</i>	Alain BELLEDENT	Directeur de Production Europe (Groupe Autoliv)

Mis en page avec la classe thloria.

À Isabelle, Nolwenn et Dylann.

*"Juger est quelquefois un plaisir
comprendre en est toujours un."*

Henri DE REIGNER (1864-1936)
Membre de l'Académie Française de 1911 à 1936 (fauteuil 39)

Table des matières

1	Introduction générale	9
1.1	Problématique et démarche de résolution	10
1.2	Origines industrielles des travaux	14
1.3	Contributions	15
1.3.1	Aspects scientifiques	15
1.3.2	Aspects techniques	16
1.4	Organisation du document	17
<hr/>		
I	Présentation du contexte	19
2	Problèmes généraux d'ordonnancement dans les systèmes de production	21
2.1	Introduction aux systèmes de production	22
2.1.1	Classification et typologie	23
2.1.1.1	Procédés mono-étage	23
2.1.1.2	Procédés multi-étages	24
2.1.1.3	Hierarchie des types	26
2.1.2	Conditions et contraintes d'exécution	27
2.1.3	Diversité de la production et flexibilité des procédés	29
2.2	Ordonnancement de la production	30
2.2.1	Caractéristiques des travaux et fonctions objectif	32

2.2.2	Spécification des problèmes d'ordonnancement	35
2.2.3	Méthodes de résolution	39
2.2.3.1	Méthodes exactes	40
2.2.3.2	Méthodes heuristiques	40
2.2.3.3	Règles de tri	42
2.2.4	Évaluation des solutions	45
2.3	Conclusion	45
3	Description des lignes d'assemblage considérées	47
3.1	Généralités sur les lignes d'assemblage	48
3.1.1	Historique	49
3.1.2	Modèles théoriques et problèmes d'ordonnancement classiques	49
3.2	Caractéristiques de base des lignes d'assemblage étudiées	53
3.2.1	Considérations principales	53
3.2.2	Données statiques	54
3.2.3	Données dynamiques	57
3.2.4	Évaluation de la cadence de production	57
3.2.5	Calcul du makespan	62
3.2.6	Latence et périodicité d'activité des machines	64
3.2.7	Exemples de systèmes simulés	65
3.2.7.1	Cadence fixée par la machine la plus lente	66
3.2.7.2	Cadence fixée par une phase de transfert	68
3.3	Mise en œuvre d'un contrôleur central	70
3.3.1	Notions de contrôle et de commande	70
3.3.2	Structure du système de pilotage	71
3.3.3	Charge du contrôleur et interactions avec les machines	72
3.4	Expression de la problématique	75
3.4.1	Contrainte sur le temps de réponse du contrôleur	75
3.4.2	Spécification du problème d'ordonnancement	76
3.5	Conclusion	77

II	Étude des systèmes considérés	79
4	Modélisation des lignes d'assemblage sous la forme d'un réseau de Petri	81
4.1	Modes d'abstraction des systèmes	82
4.2	Analyse de performances	84
4.2.1	Critères à analyser dans le cadre de l'étude	84
4.2.2	Sélection d'une méthode	85
4.2.3	Détail des phases de modélisation et de simulation	87
4.3	Choix d'un formalisme de modélisation	88
4.3.1	Principaux outils pour l'étude de systèmes à événements discrets	88
4.3.2	Intérêts des réseaux de Petri pour l'analyse de performances des systèmes de production	89
4.4	Description des éléments du modèle	90
4.4.1	Activité d'une machine de production	91
4.4.2	Enchaînement des machines de production	93
4.4.3	Traitement de messages de requête par le contrôleur	94
4.4.3.1	Cas non-préemptif et préemptif	94
4.4.3.2	Sélection de la machine à servir	97
4.4.4	Construction du modèle coloré temporisé	98
4.5	Définition des configurations étudiées	106
4.5.1	Paramétrage initial des machines de production	107
4.5.2	Déclinaisons de configurations pour les systèmes existants . .	109
4.5.3	Règles de tri évaluées et principal élément de comparaison . .	110
4.6	Conclusion	111
5	Présentation et synthèse des résultats de simulation	113
5.1	Choix des configurations évaluées et des résultats présentés	115
5.2	Définition et évaluation de la borne mécanique	116
5.3	Analyse de la cadence de production	117
5.3.1	Comparaison des règles de tri	117
5.3.2	Mise en évidence de l'impact du contrôleur	121

5.4	Observation du temps de réponse du contrôleur	122
5.5	Cadence imposée par le contrôleur	123
5.5.1	Formulation d'une borne de contrôle	123
5.5.2	Comparaison des bornes sur la cadence de production	125
5.5.3	Redéfinition de la condition d'optimalité et de la notion de latence	130
5.6	Extension des simulations	131
5.6.1	Choix d'une gamme opératoire et des règles de tri évaluées	131
5.6.2	Résultats d'exécution	132
5.6.3	Étude particulière de la règle de tri <i>FIFO</i>	141
5.6.4	Évaluation d'une configuration bornant la cadence de production par la machine la plus lente	144
5.7	Conclusion	147

III Résolution de la problématique 149

6 Atteinte du délai optimal d'inter-sorties de pièces 151

6.1	Sélection des facteurs d'optimalité	152
6.1.1	Respect de la périodicité d'activité des machines de production et du contrôleur central	153
6.1.2	Gestion de l'accès au contrôleur central	154
6.2	Méthode de synchronisation des machines de production	154
6.2.1	Principe général de la méthode	155
6.2.2	Représentation et permutation de la séquence opératoire d'une machine de production	156
6.2.3	Évaluation des combinaisons d'une gamme opératoire	162
6.2.4	Répartition de la fonction de coordination	167
6.2.5	Critères complémentaires pour la sélection d'une combinaison d'une gamme opératoire	168

6.2.6	Principales conséquences de la synchronisation des machines de production	169
6.3	Conclusion	170
7	Propositions pour un ordonnancement en ligne des messages de requête	171
7.1	Intérêt d'un ordonnancement en ligne	172
7.2	Choix des règles de tri utilisées	173
7.3	Particularité du contexte d'utilisation des règles de tri	176
7.3.1	Calcul des priorités	177
7.3.2	Sélection de la machine la plus prioritaire	179
7.4	Validation des règles de tri par simulation	179
7.4.1	Cas d'une configuration bornée par une machine de production	180
7.4.2	Cas d'une configuration bornée par le contrôleur central . . .	183
7.4.3	Cas d'une configuration sans possibilité d'atteinte de l'optimalité	186
7.5	Conclusion	188
<hr/>		
8	Conclusion générale	189
8.1	Bilan des travaux	190
8.2	Travaux prospectifs	193
<hr/>		
Annexes		195
A	Éléments de complexité	197
A.1	Évaluation de l'étendue de l'espace à explorer	198
A.2	Classes de problèmes	198
A.3	Expression de l'efficacité d'un algorithme de résolution	200
B	Adaptation de la méthode de Welch	203
B.1	Présentation de la méthode	204

B.1.1	Objectif	204
B.1.2	Déroulement	204
B.2	Simplification de l'usage	204
C	Recueil des résultats de simulation	205
C.1	Cadence de production	206
C.2	Temps de réponse du contrôleur	206
D	Présentation du logiciel Munster	207
D.1	Intérêt du développement	208
D.2	Architecture logicielle	209
D.3	Utilisation de l'outil	210
D.3.1	Arguments de la ligne de commande	210
D.3.2	Format des fichiers de ressources et d'opérations	211
D.4	Formes de production des résultats d'exécution	211
D.5	Évolutions en cours ou prévues	211
D.5.1	Réalisation d'une interface graphique	211
D.5.2	Analyse statistique	212
D.5.3	Export des résultats	212
D.5.4	Extension aux systèmes flow-shop hybrides	212
E	Prototypage d'un contrôleur sous Linux temps réel	213
E.1	Objectif du travail	214
E.2	Choix de la solution RTAI	214
E.3	Niveau de performances atteint	214
E.4	Bilan	214
<hr/>		
	Bibliographie	215
<hr/>		
	Symboles et notations	243

Table des figures	247
Liste des tableaux	251

Résumé	253
Abstract	255
Index	257

Chapitre 1

Introduction générale

Sommaire

1.1	Problématique et démarche de résolution	10
1.2	Origines industrielles des travaux	14
1.3	Contributions	15
1.3.1	Aspects scientifiques	15
1.3.2	Aspects techniques	16
1.4	Organisation du document	17

L'activité des entreprises industrielles s'inscrit dans un contexte concurrentiel de plus en plus prononcé et dicté par une économie de l'offre [Brilman, 1995]. Cette situation est conjuguée à l'accélération de la rapidité d'obsolescence des produits fabriqués. Les entreprises se doivent d'être attentives au maintien de leur capacité à répondre aux exigences croissantes de leur environnement économique en respectant un processus de progrès permanent [Goldratt and Cox, 1993].

Dans le cas de l'industrie manufacturière assurant une production de masse, la recherche permanente de rentabilité des procédés apparaît alors comme un objectif majeur. Les efforts se concentrent notamment sur l'amélioration de la productivité qui constitue l'un des principaux aspects du niveau de performances des systèmes de production. Le degré de productivité d'un système de production est fortement conditionné par la pertinence de la répartition et de la planification des opérations sur les différentes ressources qui le constituent [Bestwick and Lockger, 1979] [Herrmann, 2006].

Des problèmes d'ordonnancement apparaissent ainsi dans l'industrie manufacturière dès lors qu'il s'agit d'exploiter de manière efficace les systèmes de production. Leur résolution consiste à organiser la réalisation de travaux en répartissant les opérations à exécuter sur les différentes ressources disponibles en respectant des contraintes de délais. L'établissement d'un ordonnancement efficace correspond à une coordination de l'exécution des opérations en utilisant de manière performante les machines de production implantées sur le système.

Les contraintes de performances sont particulièrement fortes lorsque la production est réalisée au moyen de lignes d'assemblage. Les systèmes relevant de cette typologie de procédés sont en effet destinés à assurer la fabrication de volumes importants de produits finis. Une organisation judicieuse de l'activité de chacune des ressources représente donc un facteur majeur du niveau de performances. La résolution des problèmes d'ordonnancement dans l'exploitation de leur systèmes de production revêt ainsi une importance capitale pour les entreprises industrielles.

1.1 Problématique et démarche de résolution

Au cours des dernières décennies, l'amélioration des performances des systèmes de production s'est principalement appuyée sur un perfectionnement des équipements, tels que les actionneurs, les réseaux industriels de communication et les automates pro-

grammables, agissant directement sur l'écoulement des flux de matières. Elle a essentiellement été obtenue par l'utilisation de techniques issues des domaines de l'automatique et de la productique.

Cependant, le niveau croissant d'automatisation pour répondre aux exigences de réactivité et de flexibilité des systèmes de production modernes mais aussi de traçabilité des produits fabriqués s'accompagne d'une augmentation permanente du volume d'informations traitées. Les besoins concernent notamment des fonctionnalités de supervision, de contrôle et de coordination des machines en temps réel imposant la mise en œuvre de moyens de communication efficaces entre les différentes ressources.

Il n'est alors plus possible de ne considérer que les seuls aspects mécaniques d'un système de production lors de la phase de conception. La prise en compte et l'intégration des contraintes et de l'impact liés au système de pilotage devient nécessaire [Denis and Lesage, 1995]. En effet, le comportement et le niveau de performances d'un système de production sont aussi conditionnés par les caractéristiques de des équipements qui en assurent le pilotage [Grieco et al., 2001].

Les systèmes de production présentent une structure de plus en plus complexe. Cette évolution rend difficile leur évaluation et leur validation durant la phase de conception à l'aide de méthodes analytiques en raison de la nécessité de prendre en compte un grand nombre d'activités réalisées par les différents équipements qui les composent et d'interactions entre ces ressources [El-Tamini et al., 1989]. La simulation est alors devenue l'un des outils privilégiés des concepteurs. Elle permet de constater les effets des choix d'architecture matérielle et logicielle sur les performances des systèmes pilotés [Meunier and Denis, 1997]. En effet, la simulation d'un système de production revient à analyser à la fois son processus de fabrication, regroupant les activités de nature physique, et son processus de pilotage correspondant aux activités de nature logique et décisionnelle [Habchi, 2001].

Les procédés que nous considérons dans le cadre de cette thèse n'échappent pas à cette problématique d'analyse et de conception. Ils correspondent à des lignes d'assemblage automatisées pouvant être constituées de plusieurs dizaines de machines de production. Leur particularité tient dans le fait que la coordination des machines, et donc de la partie opérative, est assurée par un contrôleur central. Selon une description courante d'organisation hiérarchisée des fonctions de pilotage des systèmes de production manufacturière, les fonctions conjuguées de conduite et de commande sont con-

cernées [Trentesaux and Sénéchal, 2002]. De multiples messages sont alors échangés entre les machines et le contrôleur tout au long de la phase de production. Ils ont pour rôle de transporter les informations permettant au système d'exécuter son processus de fabrication selon des conditions préalablement définies et de répondre à des impératifs de traçabilité.

Le contrôleur central est une ressource partagée par l'ensemble des machines de production. Il réagit par l'émission d'une réponse à chacune des requêtes envoyées par les machines au cours du déroulement de leur cycle opératoire. Nos travaux s'appliquent à l'organisation de l'activité du contrôleur central, ce qui revient à ordonnancer une ressource partagée. Bien que de nombreuses études aient été publiées sur le sujet [Tantawy et al., 1992] [Saraph, 2003] [Li et al., 2004], aucune ne concerne précisément le cas que nous traitons. Elles ne sont pas en rapport avec une problématique relevant de systèmes manufacturiers organisés en ligne et intégrant un grand nombre de machines de production sollicitant un unique contrôleur dont l'activité est celle d'une ressource assurant un traitement unitaire des travaux.

L'activité d'une machine de production des systèmes que nous considérons se déroule indépendamment des autres machines dès lors qu'une pièce est disponible à son entrée. Chacune des machines répète ainsi, pour le traitement de chaque pièce à produire, un cycle opératoire dédié.

Une étude basée sur des phases de modélisation puis de simulation est nécessaire afin, d'une part, d'appréhender le comportement dynamique global des systèmes de production considérés et, d'autre part, d'évaluer leurs performances. Elle a pour but de dégager les propriétés essentielles des lignes d'assemblage contrôlées et de mettre en évidence les effets de la fonction de coordination assurée par un contrôleur central partagé par l'ensemble des machines. L'analyse des données collectées lors de la simulation nous permet également de confronter les résultats obtenus à ceux que procureraient les mêmes systèmes dépourvus de contrôleur ou pour lesquels la durée de traitement des messages serait nulle. Les informations recueillies nous apportent ainsi un certain niveau de connaissances des systèmes étudiés en permettant, notamment, d'identifier les ressources critiques constituant les goulets d'étranglement dont l'effet est de borner la cadence de production.

Les objectifs visés par nos travaux sont avant tout orientés vers la minimisation du délai d'inter-sorties de pièces, et donc de la maximisation de la cadence de production.

Nous proposons alors la définition suivante pour le problème principalement traité dans le cadre des travaux présentés dans ce document :

Définition du problème : *Étant donné un ensemble \mathcal{J} de n travaux à réaliser sur un système de production en ligne constitué d'un ensemble \mathcal{M} de k ressources successivement implantées et d'une ressource partagée assurant respectivement l'exécution d'opérations locales et distantes. Le temps d'exécution des opérations nécessaires au traitement d'un travail J_j sur une machine M_i est noté θ_i , avec $J_j \in \mathcal{J}$, $M_i \in \mathcal{M}$ et $\theta_i \in \mathbb{N}^*$. Certaines opérations étant confiées à la ressource partagée, trouver un ordonnancement des opérations distantes de telle sorte que le délai d'inter-sorties de pièces du système de production soit minimisé.*

D'autres critères de performances plus ou moins importants sont aussi à considérer dans l'évaluation d'un ordonnancement. Il s'agit principalement de la limitation des pièces en situation d'attente à l'entrée d'une machine et de la date d'établissement du régime stationnaire marquant la fin de montée en cadence du système et la stabilisation de l'activité des machines selon un comportement périodique. Nous prenons en compte certains de ces critères secondaires lors de l'évaluation des solutions que nous proposons.

Le respect des contraintes de précédence/antériorité existantes entre les opérations représentent un paramètre élémentaire de la qualité d'un ordonnancement. Cependant, d'autres contraintes nécessitent souvent d'être prises en compte [Grabot, 2006] :

- la synchronisation du début d'exécution de plusieurs opérations,
- la synchronisation de la fin d'exécution de plusieurs opérations,
- la synchronisation du début d'exécution d'une opération avec la fin d'exécution d'une autre,
- le respect d'une durée minimale, maximale ou exacte entre la fin d'exécution d'une opération et la début d'exécution d'une autre,
- etc.

Concernant la résolution de la problématique posée et bien que la réduction de la durée de traitement des messages apparaisse comme un axe de solution évident, nous ciblons en priorité la recherche d'un ordonnancement des messages traités par le contrôleur central. Notre motivation est basée sur une généralisation des résultats à tout système de production en ligne ayant à mettre en œuvre une ressource partagée quelle que soit la nature physique ou logique et décisionnelle de l'activité qu'elle assure.

Cependant, nous n'avons pas totalement exclu de l'objectif visé par nos travaux les améliorations de performances atteignables par la réduction de la durée de traitement des messages envoyés par les machines de production. L'abaissement de la charge d'activité confiée au contrôleur central constitue en effet un moyen simple et efficace aboutissant à un gain de productivité des systèmes. Un prototype de contrôleur a ainsi été conçu puis réalisé afin de montrer les gains obtenus par cette méthode.

1.2 Origines industrielles des travaux

Nos travaux trouvent leurs origines au sein de l'industrie manufacturière de masse et plus particulièrement dans les contraintes rencontrées par les équipementiers automobiles dans la conception, la réalisation et la gestion de leurs lignes d'assemblage. Les systèmes de production initialement étudiés sont des procédés de taille industrielle dont l'implantation et l'utilisation sont effectives dans les ateliers de la société Livbag du groupe Autoliv¹.

Autoliv est un groupe suédois dont l'activité se concentre sur le développement, le test et la production de systèmes de protection des occupants de véhicules automobiles [Nilsson, 2003]. Il est leader mondial sur le marché des équipements de sécurité automobile et emploie plus de 45 000 personnes dans une trentaine de pays. Sa filiale Livbag, située en France, est spécialisée dans la conception et la fabrication de générateurs de gaz destinés au gonflage de cousins d'air de sécurité embarqués dans l'habitacle de voitures.

Les lignes d'assemblage déployées dans les ateliers de la société Livbag ne présentent pas toutes le même niveau d'automatisation ni la même architecture de pilotage. Celles que nous avons retenues comme base d'étude présentent une connexion de toutes les machines au contrôleur chargé de leur coordination. Selon les configurations adoptées sur ces procédés, les échanges entre une machine et le contrôleur s'opèrent au début et au terme de chaque cycle opératoire.

Les procédés de ce type ont pour but de permettre l'assemblage en très grande quantité de pièces variées mais présentant de fortes similitudes et appartenant ainsi à une même famille de produits. Ces lignes d'assemblage répondent aux besoins d'une

¹<http://www.autoliv.com>

production de masse en apportant un compromis entre les notions de flexibilité et de productivité.

Les systèmes de production à partir desquels ont été initiés nos travaux représentent une base d'étude intéressante en raison des dysfonctionnements constatés lors de simples observations visuelles menées au cours de leur exécution. Les anomalies alors identifiées concernent notamment la variation du temps de réponse du contrôleur central et la manifestation de situations de suspension plus ou moins longue de l'activité des machines au cours du déroulement de leur cycle opératoire. De plus, les délais d'inter-sorties de pièces ne respectent pas toujours une période bien définie mais engendrent fréquemment des sorties de pièces en rafales.

1.3 Contributions

La démarche d'analyse et de résolution que nous avons suivie nous permet, au-delà de la seule réponse à la problématique posée, de dégager un certain nombre de contributions pouvant être classées en fonction de leur nature scientifique ou technique. Nous dressons dans cette section un inventaire des apports que constituent nos travaux.

1.3.1 Aspects scientifiques

La construction d'un modèle en vue de son exécution correspond à une part importante des travaux décrits dans cette thèse et de la démarche de résolution adoptée. Nous avons modélisé les lignes d'assemblage selon le formalisme des réseaux de Petri avec pour objectif de produire un modèle alliant des caractéristiques de compacité et de généralité. Le modèle obtenu présente en effet un haut niveau de paramétrage permettant sa réutilisation pour l'étude de systèmes de production en ligne, contrôlés ou pas, de différentes tailles. Les éléments de configuration s'appliquent principalement au nombre de machines implantées sur le système, à la définition de la séquence opératoire affectée à chaque machine ainsi qu'au délai de transfert et à la capacité de stockage de pièces entre deux machines successives. Bien que le travail abordé dans ce document se limite à la prise en compte d'un contrôleur ne servant qu'une seule machine à chaque instant, le nombre de processeurs disponibles pour assurer le traitement des messages de requête est aussi paramétrable. Le modèle peut ainsi apporter une aide à

la conception de systèmes de production en ligne en permettant une évaluation et une validation a priori de l'organisation des procédés. Certaines caractéristiques, telles que le délai d'inter-sorties de pièces, la position des ressources critiques ou les volumes de pièces en attente à l'entrée d'une machine du production, peuvent être plus particulièrement étudiées.

Les résultats de simulation démontrent l'existence d'une borne de contrôle limitant la cadence de production des systèmes d'assemblage que nous considérons. Ils nous permettent de généraliser le calcul du délai minimum d'inter-sorties des pièces, correspondant à la cadence de production optimale, pour un système de production en ligne quelle que soit la nature de la ressource critique.

Une méthode originale d'évaluation des possibilités d'atteindre la cadence optimale par un système de production en ligne contrôlé est alors présentée. Lorsque le délai minimum d'inter-sorties de pièces ne peut être atteint, cette méthode permet toutefois de prédire l'écart à l'optimalité.

Nous proposons également une discipline spécifique de consommation des messages de requête par le contrôleur central. Elle est basée sur un mécanisme d'optimisation des dates d'arrivée et des délais de rétention des pièces sur les différentes machines et vise ainsi une amélioration de l'écoulement du flux de matière à travers le système.

Les résultats obtenus restent applicables, sous certaines conditions, à tout système de production en ligne mettant en œuvre une ressource partagée par les machines. À titre d'exemple, la méthode d'ordonnancement proposée peut, dans certains cas, être utilisée pour organiser l'activité mécanique d'une ressource telle qu'un bras manipulateur ou un robot de soudure.

1.3.2 Aspects techniques

Alors qu'une réduction de la durée de traitement des messages de requête reste une réponse évidente à la problématique initialement posée, elle impose une refonte importante, voire un changement complet, du logiciel implanté sur le contrôleur des lignes d'assemblage à l'origine de nos travaux. L'amélioration du niveau de performances des systèmes concernés par une méthode d'ordonnancement des messages de requête présente l'avantage d'être applicable aux contrôleurs existants et se résume à une adaptation du mode de consommation. Cependant, nous proposons une architecture logicielle

adaptée à l'ensemble des fonctionnalités que doit assurer le contrôleur central. Les résultats obtenus avec le prototype développé montrent que les temps de traitement des messages de requête peut être considérablement réduit et rendu déterministe.

Enfin, un outil logiciel destiné à déterminer l'activité globale d'un système donné à été développé. Son principe algorithmique repose sur le calcul des dates de début, de suspension, de reprise et de fin des cycles opératoires des machines au cours de l'activité du système. Les résultats d'une exécution sont exprimés sous la forme d'un diagramme de Gantt. Le paramétrage de l'outil concerne essentiellement le nombre et la configuration des machines, la répartition de la gamme opératoire ainsi que la sélection d'une discipline de consommation des messages de requête par le contrôleur central. Les diagrammes de Gantt figurant dans la suite du document sont générés à l'aide de cet outil.

1.4 Organisation du document

Le document se décompose essentiellement en trois parties qui peuvent, dans une certaine mesure, être abordées de manière indépendante :

- exposition du contexte général des travaux,
- étude des lignes d'assemblage contrôlées,
- réponse à la problématique.

La première partie présente au deuxième chapitre une étude bibliographique des problèmes d'ordonnancement associés à la mise en œuvre de systèmes de production. Elle précise dans le troisième chapitre la structure des lignes d'assemblage étudiées et exprime les principales caractéristiques comportementales de ces systèmes de production.

L'étude par phases successives de modélisation et de simulation des procédés contrôlés que nous considérons est détaillée dans la deuxième partie du document. La construction progressive du modèle est expliquée par le quatrième chapitre alors que la synthèse des résultats obtenus lors de la phase de simulation sont exposés dans le cinquième chapitre.

La troisième partie présente les éléments de réponse que nous apportons à la problématique posée. Le sixième chapitre décrit les conditions d'atteignabilité de la cadence optimale et détaille une méthode d'évaluation des possibilités d'atteinte du délai d'inter-sorties minimum basée sur la synchronisation des machines de production. Le septième

chapitre propose un algorithme d'ordonnancement de l'activité du contrôleur central fondé sur les conditions d'atteignabilité de la cadence optimale et donc du délai d'inter-sorties minimal de pièces.

Une conclusion générale dresse un bilan de la thèse et ouvre des perspectives de poursuite des travaux.

Quelques annexes complètent le document. Elles apportent des éclaircissements quant à certains aspects des travaux ou des précisions au sujet des développements logiciels réalisés.

Partie I

Présentation du contexte

Chapitre 2

Problèmes généraux d'ordonnancement dans les systèmes de production

Sommaire

2.1	Introduction aux systèmes de production	22
2.1.1	Classification et typologie	23
2.1.1.1	Procédés mono-étage	23
2.1.1.2	Procédés multi-étages	24
2.1.1.3	Hierarchie des types	26
2.1.2	Conditions et contraintes d'exécution	27
2.1.3	Diversité de la production et flexibilité des procédés	29
2.2	Ordonnancement de la production	30
2.2.1	Caractéristiques des travaux et fonctions objectif	32
2.2.2	Spécification des problèmes d'ordonnancement	35
2.2.3	Méthodes de résolution	39
2.2.3.1	Méthodes exactes	40
2.2.3.2	Méthodes heuristiques	40
2.2.3.3	Règles de tri	42
2.2.4	Évaluation des solutions	45
2.3	Conclusion	45

L'ordonnancement est une discipline mathématique très étudiée depuis plusieurs décennies. Les problèmes d'ordonnancement se rencontrent dans une large variété de domaines et notamment dans les ateliers industriels. Dans ce contexte, ils sont généralement abordés et étudiés à des fins d'amélioration de la performance des systèmes de production.

L'objet de ce chapitre est de présenter les aspects fondamentaux concernant les problèmes d'ordonnancement rencontrés dans la mise en œuvre de systèmes de production. Nous commençons par décrire et classer les systèmes de production selon leur structure et le mode de circulation qu'adoptent les pièces à fabriquer. Les principales conditions et contraintes d'exécution considérées au cours de leur activité sont ensuite présentées ainsi que la notion de flexibilité attendue des procédés modernes.

Nous définissons ensuite ce qu'est concrètement un problème d'ordonnancement, notamment dans le cadre d'un système de production, et décrivons une manière courante de l'exprimer. Enfin, nous présentons une classification des ordonnancements pour ensuite aborder les méthodes de résolution des problèmes tout en apportant les notions fondamentales nécessaires à l'évaluation de leur complexité algorithmique.

2.1 Introduction aux systèmes de production

Un système de production est souvent défini comme un ensemble de moyens de production dans lequel sont introduits des produits bruts et duquel ressortent ces mêmes produits bruts nantis d'une valeur ajoutée [Gendreau, 1991]. Ce sont des équipements destinés à l'exécution de processus sous la forme d'une séquence d'opérations assurées par des machines dans le but de réaliser des travaux. L'opération est ainsi une tâche élémentaire dont le traitement par une ressource nécessite un certain nombre d'unités de temps.

La séquence d'opérations à respecter pour la réalisation d'un travail est couramment désignée par les termes *gamme*, *gamme opératoire* ou encore *gamme de fabrication*. Toutefois, une définition plus complète d'une gamme la présente comme une description structurée du processus de fabrication incluant des données plus exhaustives que la seule séquence d'opérations [Colin, 2003]. Dans la suite de ce document nous considérons qu'une gamme se limite à une succession ordonnée d'opérations établie pour aboutir à la réalisation d'un travail.

Les systèmes de production intègrent également des phases de transfert des pièces à fabriquer entre les machines lorsque la réalisation d'un travail impose une répartition des opérations sur plusieurs machines.

La production d'une pièce correspond ainsi à la réalisation d'un travail et les machines représentent les ressources dont dispose le système pour assurer l'exécution des opérations.

2.1.1 Classification et typologie

Il existe dans la littérature une abondante bibliographie présentant une classification des types de systèmes de production [French, 1982] [Carlier and Chétienne, 1988] [Mac Carthy and Liu, 1993] [Little and Hemmings, 1994]. Ces derniers sont généralement classés en fonction de la configuration des machines implantées et du cheminement des pièces constituant les travaux à réaliser. Cinq types principaux sont alors distingués parmi les publications traitant de problèmes d'ordonnancement dans les systèmes de production.

Les systèmes de production diffèrent par leur organisation, la nature des contraintes de précedence entre les diverses opérations nécessaires à la réalisation d'un même travail et donc par le mode de circulation des pièces sur les machines. Ils sont regroupés en deux grandes catégories selon que la production qu'ils assurent nécessite le traitement des pièces sur une ou plusieurs machines. Le terme *étage* est évoqué pour décrire un groupe de machines couplées en parallèle que visitent les pièces au cours de leur cheminement à travers le système de production.

2.1.1.1 Procédés mono-étage

Les procédés mono-étage sont les types les plus simples. Ils concernent les systèmes pourvus d'une machine unique et ceux constitués d'une batterie de machines couplées en parallèle. Dans ce second cas, les machines sont interchangeables, elles assurent toutes l'exécution de la même séquence d'opérations. Elles ne sont pas pour autant obligatoirement identiques et peuvent présenter des durées différentes d'exécution des opérations de la gamme de fabrication.

Ces types simples se limitent à un seul étage de machines et la réalisation complète d'un travail se résume alors au traitement de la pièce associée par une seule machine.

2.1.1.2 Procédés multi-étages

Pour la plupart des systèmes de production industriels, la réalisation des travaux nécessite d'être répartie sur plusieurs machines dans un ordre donné. Chaque machine n'exécute alors qu'une partie de la gamme de fabrication. Ces procédés multi-étages ont été très largement étudiés et des ouvrages de haut niveau lui ont été consacrés [Tanaev et al., 1994].

Les systèmes multi-étages font appel à des configurations étendues et organisées autour d'une série de machines. La réalisation d'un travail s'effectue au cours de passages successifs de la pièce sur plusieurs machines.

Ainsi, lorsque les opérations sont indépendantes, il s'agit d'un *open-shop*. Les pièces visitent alors les machines dans un ordre totalement libre car les opérations nécessaires à la réalisation d'un travail ne sont pas affectées par une contrainte de précédence [Gonzalez and Sahni, 1976] [Kubiak et al., 1991].

Lorsque l'ordre des opérations d'un travail n'est pas nécessairement identique à celui de l'ensemble des autres travaux, le type *job-shop* est évoqué. Chaque travail fait l'objet d'une séquence dédiée d'exécution des opérations et d'un ordre de passage propre sur les machines du système de production [Mellor, 1966] [Blazewicz, 1987]. La figure 2.1 présente un exemple de système *job-shop* constitué de trois machines, désignées par M_1 , M_2 et M_3 , et réalisant trois travaux notés J_1 , J_2 et J_3 . Les arcs représentent le cheminement suivi par les pièces lors de leur fabrication.

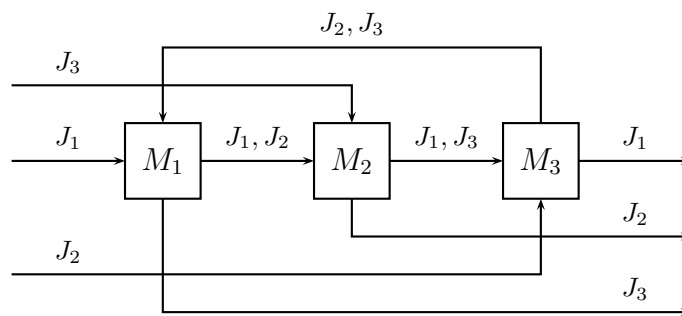


FIG. 2.1: *Système job-shop*

Quand les machines sont organisées en série et que l'enchaînement des opérations des travaux respecte toujours le même ordre, le système de production est défini par le terme *flow-shop* [Gupta, 1979] [Dudek et al., 1991]. La séquence opératoire est identique

pour tous les travaux à réaliser et une contrainte implicite de précedence s'applique aux opérations. La figure 2.2 montre un exemple de système flow-shop.

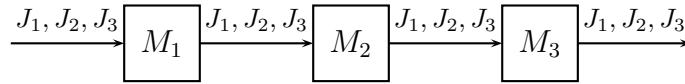


FIG. 2.2: *Système flow-shop*

Les types multi-étages rencontrés dans l'industrie font parfois l'objet d'une configuration plus complexe intégrant au niveau d'un même étage du système de production des groupes de machines couplées en parallèle. Les machines d'un étage sont toutes affectées à l'exécution du même jeu d'opérations. Ces structures sont appelées *open-shop généralisé*, *job-shop généralisé* et *flow-shop hybride*. Ce sont des généralisations des procédés multi-étages classiques qui ne disposent que d'une seule machine à chaque étage. L'étude des systèmes de production généralisés peut être abordée en considérant une homogénéité ou une hétérogénéité des machines parallèles. La figure 2.3 illustre un exemple de réalisation de trois travaux J_1 , J_2 et J_3 sur un système flow-shop hybride à trois étages. Le deuxième et le troisième étages sont respectivement constitués de trois et de deux machines couplées en parallèle.

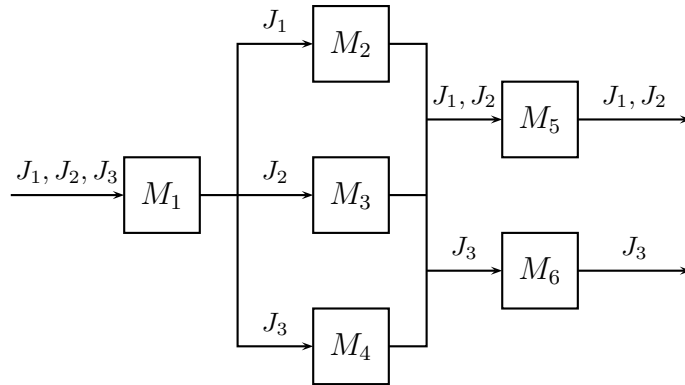


FIG. 2.3: *Système flow-shop hybride*

Plus particulièrement pour les systèmes flow-shop, bien qu'ils aient fait l'objet de nombreuses études sous leur forme hybride [Linn and Zhang, 1999] [Vignier et al., 1999], d'autres types peuvent être déclinés en fonction de la complexité et de la spécificité de l'organisation des machines [Telle, 2003].

Il existe en outre des équipements de production au sein desquels le cheminement est imposé à certaines pièces alors qu'il est libre pour d'autres [Masuda et al., 1985] [Shakhlevich et al., 1999]. L'étude d'un tel modèle repose sur le principe que les systèmes de production rencontrés dans les ateliers industriels ne sont pas toujours de purs procédés job-shop ou flow-shop [Ferrell et al., 2000]. Ces systèmes mixtes sont couramment identifiés par le terme *mixed-shop*. Ils peuvent être considérés comme une généralisation de tous les autres types de procédés.

2.1.1.3 Hiérarchie des types

En synthèse, nous présentons une classification des différentes configurations de systèmes de production sous la forme d'une hiérarchie des modèles en tenant compte de critères d'appartenance d'un type à un autre. La figure 2.4 montre la hiérarchie des types de systèmes de production. Lorsqu'un type peut être décliné à partir d'un autre, l'arc que les relie pointe sur le type généralisé.

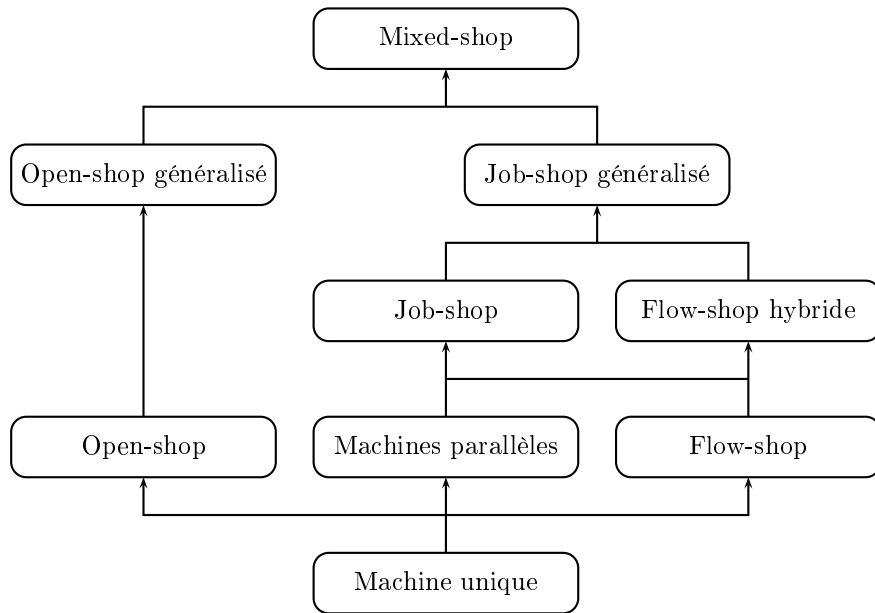


FIG. 2.4: Hiérarchie des types de systèmes de production

Par exemple, le flow-shop simple est un cas particulier du flow-shop hybride. De même, un procédé constitué de machines parallèles est une généralisation d'un système à machine unique.

2.1.2 Conditions et contraintes d'exécution

Un grand nombre de conditions ou de contraintes d'exécution peuvent être considérées lors de l'étude d'un système de production. Ces règles régissent l'activité des machines et le comportement global des procédés modélisés. Nous présentons ici les plus fréquemment rencontrées dans la littérature.

Le plus souvent, les conditions et contraintes d'exécution s'appliquent directement aux machines. Cependant, elles peuvent aussi se rapporter aux opérations ou aux travaux. En effet, une contrainte couramment retenue dans les problèmes d'ordonnancement concerne la prise en compte de dates de début ou d'échéance pour l'exécution d'une opération ou la réalisation d'un travail. Ces dates peuvent être impératives ou souhaitées. La contrainte est alors basée sur une *échéance stricte* ou une *échéance souple*.

Une contrainte d'exécution basée sur des délais à respecter entre deux opérations successives d'un même travail a également été étudiée. L'exécution d'une opération donnée ne peut ainsi démarrer qu'après qu'un laps de temps se soit écoulé depuis la fin d'exécution de l'opération précédente [Fondrevelle, 2005].

Quelques travaux ont également portés sur l'étude de systèmes au sein desquels la durée d'exécution des opérations est variable [Jiang, 1997].

L'un des aspects importants des systèmes de production réside dans la possibilité accordée aux machines de suspendre l'exécution d'une opération au profit d'une autre jugée plus prioritaire. Dans le cas où les tâches sont interruptibles, le procédé adopte un mécanisme dit de *préemption* [Berry, 1993].

La recherche d'une solution à un problème d'ordonnancement peut aussi prendre en compte la possibilité de laisser inactive une machine alors qu'une pièce est disponible à son entrée. Ce type d'ordonnancement, parfois qualifié d'*ordonnancement oisif*, repose sur une intégration de temps creux dans l'activité des machines [Lebre, 2006].

L'analyse théorique d'un système de production est souvent abordée en considérant une capacité infinie de stockage de pièces entre deux machines successives. Le terme *encours* est utilisé pour décrire la situation des pièces stationnant entre deux machines. Toutefois, dans le cas d'études basées sur des applications industrielles concrètes, il est parfois nécessaire d'intégrer aux modèles une contrainte limitant la taille d'encours [Leisten, 1990] [Brucker et al., 2003]. La saturation d'un tel espace tampon impose l'arrêt d'activité de la machine située en amont.

Les procédés sans attente, également désignés par le terme anglais *no-wait systems*, sont des systèmes au sein desquels les pièces circulent sans marquer de temps d'arrêt. Les phases observées ne sont liées qu'à la production ou l'acheminement et les pièces ne sont jamais en situation d'encours [Bouquard et al., 2005] [Liaw et al., 2005] [Spieksma and Woeginger, 2005]. Concernant plus précisément le cas des systèmes de type flow-shop, il a été montré que la problématique induite est apparentée à celle du voyageur de commerce [Piehler, 1960]. Cette manière d'aborder l'analyse d'un flow-shop sans attente a largement été traitée dans la littérature [Reddi and Ramamoorthy, 1972] [Emmons and Mathur, 1995].

Les systèmes de production sont aussi parfois étudiés selon une contrainte d'activité permanente des machines durant toute la durée d'exécution de l'équipement. Ces procédés sans temps d'arrêt, *no-idle systems* en anglais, représentent des systèmes nécessitant un fonctionnement ininterrompu des machines. Dès lors qu'une machine a commencé la réalisation du premier travail, elle ne doit marquer aucun temps d'arrêt avant d'avoir achevé le dernier travail [Saadani, 2003] [Kamburowski, 2004].

Une contrainte d'utilisation des machines de production peut également être prise en compte lors de l'étude d'un équipement de production. Certaines machines peuvent ne disposer que d'une disponibilité partielle au cours de la période d'exécution du système [Aggoune, 2003] [Gharbi and Haouari, 2005] [Breit, 2004]. Une telle considération est retenue pour intégrer au modèle des actions de maintenance ou des pannes.

L'étude de systèmes flow-shop est souvent menée en considérant une contrainte de permutation. Elle impose que les pièces visitent les machines en respectant leur position initiale. Aucun dépassement ne peut avoir lieu au cours de l'exécution du système de production. Les travaux sont ainsi toujours réalisés selon un ordre unique par les machines [Lageweg et al., 1978] [Potts et al., 1991]. Le flow-shop de permutation constitue une modélisation typique des systèmes de production agencés en ligne.

Des contraintes de précédence entre les opérations sont aussi largement considérées [Lim and Mac Mahon, 1994] [Guinet and Legrand, 1998] [Gladky et al., 2004]. La fabrication de produits manufacturiers impose en effet souvent l'exécution des opérations de la gamme dans un ordre donné.

Enfin, il existe dans la littérature un grand nombre de publications relatives à des travaux s'appliquant à des systèmes de production intégrant une combinaison de contraintes [Espinouse et al., 1999] [Cheng and Liu, 2003] [Wang et al., 2005].

2.1.3 Diversité de la production et flexibilité des procédés

La conjoncture économique dans laquelle évoluent les entreprises industrielles impose que les systèmes de production soient, non seulement conçus pour traiter d'importants volumes de pièces, mais également pour s'adapter à plusieurs variantes de produits finis. La qualité des procédés est alors aussi évaluée en fonction de leur flexibilité, c'est-à-dire de leur aptitude à produire différents modèles de pièces et de leur rapidité à opérer les reconfigurations. Trois types de procédés sont identifiés selon la diversité de la production et de la manière d'alterner les modèles de pièces [Wild, 1972] [Buxey et al., 1973].

Lorsque les temps de configuration des machines, impliquant la plupart du temps un changement d'outillage, sont trop pénalisants pour atteindre un taux d'utilisation profitable des équipements, la production est limitée à un seul modèle de pièces. Les procédés appartenant à cette classe sont dits *systèmes à modèle unique*.

L'automatisation progressive et l'évolution générale des technologies ont permis de réduire les temps de configuration et donc leur impact sur la rentabilité des systèmes de production. Ceux-ci ont alors été conçus pour la fabrication de plusieurs modèles de pièces. Dans ce cas, l'activité des équipements est découpée en plusieurs lots de production séparés par une phase de configuration. Un seul modèle est ainsi véritablement en cours de production à chaque instant. La désignation *systèmes à modèles multiples* concerne ces procédés.

La reconfiguration des systèmes de production doit être la plus rapide possible lorsqu'il s'agit de répondre efficacement à la nécessité de changements fréquents du modèle de pièces à produire en minimisant les effets sur les cadences. Lorsque les temps de reconfiguration sont rendus négligeables par la similitude des caractéristiques des modèles et des processus de production, les équipements sont en mesure d'assurer une production simultanée de différents modèles. Les pièces ne sont alors plus regroupées en lots de production en fonction de leur modèle mais elles sont produites dans un ordre indifférent. Les machines alternent ainsi les modèles de pièces au cours d'un même lot de production. Ce type d'équipements, désigné par le terme *systèmes à modèles mélangés*, présente un niveau maximal de flexibilité.

Les systèmes à modèles multiples et mélangés sont souvent considérés comme appartenant à la même classe. Le second étant un cas particulier du premier pour lequel la taille des lots de production est fixée à une pièce.

La figure 2.5 montre un exemple comparatif des trois types d'enchaînement des modèles de pièces sur un système de production. les différents modèles sont distingués par la couleur ou le motif de remplissage des carrés symbolisant les pièces produites.

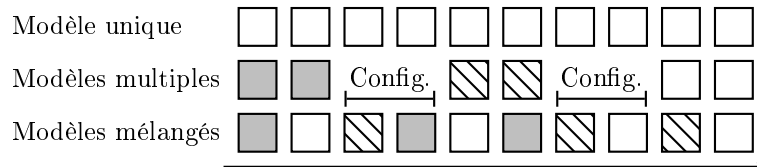


FIG. 2.5: *Enchaînement des productions*

La flexibilité apparaît comme une caractéristique majeure des systèmes de production modernes. Elle s'accompagne d'une réduction des tâches manuelles et fait largement appel à des moyens informatisés ou robotisés car elle nécessite un niveau élevé d'automatisation. De nombreuses publications lui ont été consacrées depuis une vingtaine d'années [Brown et al., 1984] [Harley, 1984] [Sethi and Sethi, 1990] [Widmer, 1991].

2.2 Ordonnancement de la production

Dans le cadre d'un système de production, la répartition des opérations sur les machines impose de prendre en compte la structure du procédé et un certain nombre de contraintes agissant sur les opérations ou les machines. Il est alors nécessaire de résoudre un problème d'ordonnancement afin de permettre une exécution conforme du système de production tout en respectant les contraintes.

Un problème d'ordonnancement se compose d'un ensemble de tâches dont l'exécution est assurée par des ressources dans un environnement contraint. Un ordonnancement a pour but de définir les dates de début d'exécution des tâches en considérant la disponibilité des ressources et en respectant les contraintes imposées avec pour objectif d'optimiser un ou plusieurs critères. Deux hypothèses importantes sont communément retenues dans les problèmes d'ordonnancement classiques. Il est en effet généralement considéré qu'une ressource ne peut exécuter qu'une seule tâche à chaque instant et qu'une tâche ne peut pas être exécutée simultanément par deux ressources.

Les problèmes d'ordonnancement et leurs applications sont extrêmement variés et ont fait l'objet d'un grand nombre d'ouvrages et publications [Chu and Proth, 1996]

[Esquirol and Lopez, 1999] [Lopez and Roubellat, 2001]. Ils sont notamment rencontrés dans la planification de projets, la gestion de ressources humaines, l'exécution de programmes informatiques et la mise en œuvre de systèmes de production. Dans ce dernier cas, les problèmes traités sont classés dans une catégorie appelée *ordonnancement d'atelier*. Les problèmes d'ordonnancement sont scientifiquement abordés depuis les années 50. Les premiers cas étudiés étaient surtout orientés vers des problèmes liés à la performance de systèmes de production [Johnson, 1954] [Jackson, 1955].

Les problèmes d'ordonnancement d'atelier font appel à des ressources *renouvelables*. En effet, après leur utilisation pour l'exécution d'une ou plusieurs opérations, les machines sont à nouveau disponibles en même quantité. Ces ressources sont aussi *disjonctives*, c'est-à-dire donc non partageables, dans la mesure où elles ne peuvent assurer qu'une seule exécution d'opération à la fois. La figure 2.6 expose le positionnement par les ressources des problèmes d'ordonnancement d'atelier.

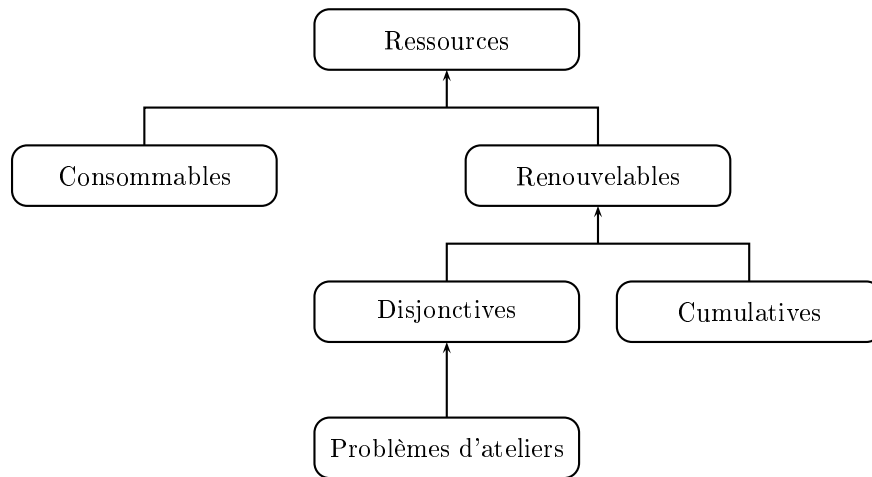


FIG. 2.6: *Positionnement des problèmes d'ordonnancement d'atelier*

Dans un système de production, une tâche est souvent un élément d'une séquence d'opérations destinée à la réalisation d'un travail. Par conséquent, une production correspond à un ensemble $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$ de n travaux dont la réalisation est assurée par un ensemble $\mathcal{M} = \{M_1, M_2, \dots, M_k\}$ de k machines et concrétisée par la fabrication d'un lot de n pièces. Chaque travail J_j est réalisé par les exécutions successives d'un ensemble $\mathcal{O}_j = \{O_{1,j}, O_{2,j}, \dots, O_{m,j}\}$ de m opérations. Le couple i, j d'une opération $O_{i,j}$ indique alors qu'il s'agit de la $i^{\text{ème}}$ opération de la séquence nécessaire à la

réalisation du travail J_j . Il est généralement admis que le nombre m d'opérations est égal au nombre d'étages du système.

Le problème d'ordonnancement dans les systèmes de production consiste alors à organiser dans le temps l'exécution d'opérations interdépendantes à l'aide des ressources disponibles en quantité limitée pour réaliser un plan de production [Erschler et al., 1992].

2.2.1 Caractéristiques des travaux et fonctions objectif

Dans le contexte de l'ordonnancement de tâches dans les systèmes informatiques temps réel, un modèle canonique est employé pour représenter les principaux paramètres temporels des tâches. Dans ce modèle, une tâche est définie à l'aide de caractéristiques chronologiques et chronométriques [Cottet et al., 2000]. Les premières concernent des dates alors que les secondes s'appliquent à des délais. Il faut aussi discerner, parmi les caractéristiques, les données statiques et dynamiques. Les caractéristiques statiques sont les paramètres de base d'une tâche alors que les caractéristiques dynamiques représentent les données résultant de l'exécution du système. Les travaux réalisés par les machines d'un système de production peuvent être modélisés de manière analogue.

Les caractéristiques statiques essentielles d'un travail J_j sont les suivantes :

- la date de disponibilité r_j ,
- la date d'échéance d_j ,
- la durée d'exécution p_j ,
- la pondération w_j .

La date de disponibilité correspond à l'instant à partir duquel le travail peut commencer. Il s'agit donc de la date de démarrage au plus tôt du travail. La date d'échéance indique l'instant de fin de réalisation souhaitée pour le travail. La durée d'exécution se rapporte à cumul des temps opératoires effectivement nécessaires pour la réalisation du travail et ne tient pas compte des temps d'attente. La pondération est destinée à modéliser des notions de coût ou d'importance dans la réalisation de travaux.

Les principales caractéristiques dynamiques sont indiquées ci-dessous :

- la date de début d'exécution t_j ,
- la date d'achèvement c_j ,
- la durée de séjour sur le système $F_j = c_j - r_j$,
- le retard algébrique $L_j = c_j - d_j$,

- le retard absolu $T_j = \max\{0, c_j - d_j\}$,
- l'avance $E_j = \max\{0, d_j - c_j\}$,
- l'indicateur de retard $U_j = \{1 \text{ si } T_j > 0, 0 \text{ sinon}\}$.

Les données statiques et dynamiques peuvent s'appliquer identiquement à une opération O_{ij} d'un travail J_j sur une machine M_i . Il est en effet parfois utile de préciser certaines caractéristiques au niveau des opérations ou d'évaluer le comportement d'un système en analysant les dates et les durées d'exécution des différentes opérations.

La figure 2.7 présente un diagramme temporel de la réalisation sur un système donné de deux travaux composés d'une seule opération. Elle montre un premier travail J_j qui respecte son échéance et un travail suivant J_{j+1} qui s'achève en retard.

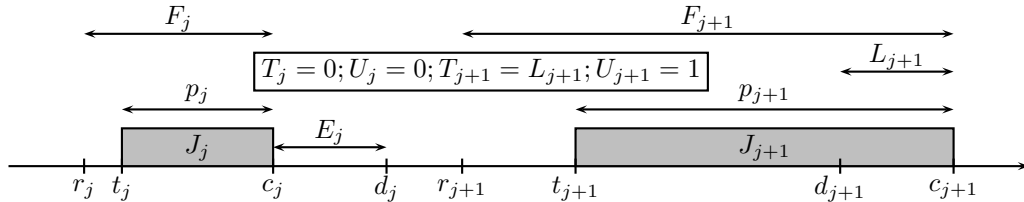


FIG. 2.7: Diagramme temporel de réalisation de travaux

L'étude théorique de problèmes d'ordonnancement appliqués à des systèmes de production considère parfois que l'ensemble des travaux est disponible à l'initialisation du système. Dans ce cas particulier, nous avons $\forall J_j \in \mathcal{J} : r_j = 0$ et donc $F_j = c_j$. Cette hypothèse n'est pas toujours adaptée à l'analyse de problèmes pratiques.

L'objectif d'un ordonnancement est de minimiser ou maximiser un critère d'évaluation numérique donné en respectant un certain nombre de contraintes. De nombreuses publications s'appliquent aussi à des problèmes multi-objectifs [Ramesh and Cary, 1989] [Hoogeveen, 2003] [T'kindt et al., 2003]. Les fonctions objectif classiquement visées permettent l'évaluation globale de la qualité d'un ordonnancement basée sur des délais. L'optimisation peut concerner de nombreux critères. Elle peut ainsi s'attacher à minimiser la durée de fabrication d'un lot de pièces ou les retards. La date d'achèvement du dernier travail, notée C_{max} , représente le critère le plus souvent visé lors de l'évaluation de performance d'un ordonnancement appliqué à un système de production. Elle est généralement désignée par le terme anglais *makespan* et donne le temps total d'exécution du système pour réaliser un ensemble de travaux. Le tableau 2.1 liste les principaux critères rencontrés dans la littérature.

Critère	Description
$C_{max} = \max_{J_j \in \mathcal{J}} \{c_j\}$	Date d'achèvement du dernier travail
$C_{sum} = \sum_{J_j \in \mathcal{J}} c_j$	Somme des dates d'achèvement
$F_{max} = \max_{J_j \in \mathcal{J}} \{F_j\}$	Durée maximale de séjour sur le système
$F_{avg} = \frac{1}{n} \sum_{J_j \in \mathcal{J}} F_j$	Durée moyenne de séjour sur le système
$F_{sum} = \sum_{J_j \in \mathcal{J}} F_j$	Flot total
$L_{max} = \max_{J_j \in \mathcal{J}} \{L_j\}$	Retard algébrique maximal
$T_{max} = \max_{J_j \in \mathcal{J}} \{T_j\}$	Retard absolu maximal
$T_{avg} = \frac{1}{n} \sum_{J_j \in \mathcal{J}} T_j$	Retard absolu moyen
$T_{sum} = \sum_{J_j \in \mathcal{J}} T_j$	Somme des retards absolus
$U_{sum} = \sum_{J_j \in \mathcal{J}} U_j$	Nombre de travaux en retard
$E_{max} = \max_{J_j \in \mathcal{J}} \{E_j\}$	Avance maximale
$E_{sum} = \sum_{J_j \in \mathcal{J}} E_j$	Somme des avances
$C_{sum}^w = \sum_{J_j \in \mathcal{J}} w_j c_j$	Somme pondérée des dates d'achèvement
$T_{sum}^w = \sum_{J_j \in \mathcal{J}} w_j T_j$	Somme pondérée des retards absolus
$T_{avg}^w = \frac{\sum_{J_j \in \mathcal{J}} w_j T_j}{\sum_{J_j \in \mathcal{J}} w_j}$	Retard absolu moyen pondéré
$U_{sum}^w = \sum_{J_j \in \mathcal{J}} w_j U_j$	Nombre pondéré de travaux en retard
$U_{avg}^w = \frac{\sum_{J_j \in \mathcal{J}} w_j U_j}{\sum_{J_j \in \mathcal{J}} w_j}$	Nombre moyen pondéré des travaux en retard

TAB. 2.1: Principaux critères d'optimisation

Les critères évoluant de façon non décroissante lorsque les dates d'achèvement des travaux augmentent sont considérés comme réguliers. Ils s'expriment alors comme des fonctions $f(c_1, c_2, \dots, c_n)$ des dates c_j de fin de réalisation des travaux. La figure 2.8 montre les relations élémentaires existant entre les principaux critères réguliers. L'amélioration d'un critère implique celle d'un critère dépendant et pointé par un arc de liaison.

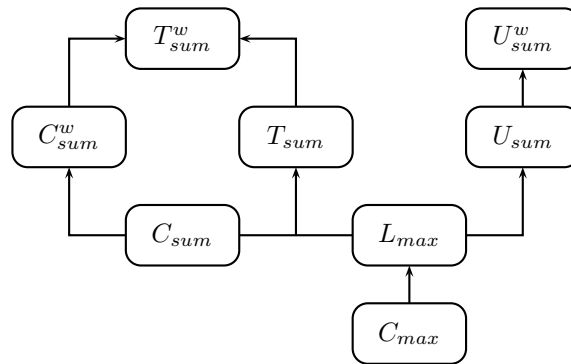


FIG. 2.8: Relations entre les principaux critères réguliers

D'autres critères peuvent également faire l'objet d'une recherche d'optimabilité. Il s'agit notamment de la minimisation des encours de travaux sur le système [Sadfi, 2002], de la maximisation de l'activité des machines ou de l'équilibrage de charge sur le système. Ce dernier cas a largement été étudié [Baybars, 1986] [Becker and Scholl, 2006] et consiste à optimiser la répartition des opérations sur les différentes machines.

Enfin, il est nécessaire de préciser que la performance des systèmes de production industriels ne se restreint pas à l'unique satisfaction de délais mais intègre aussi des critères de coût et de qualité. Dans le cadre de notre travail, nous nous limitons toutefois aux seuls aspects temporels des ordonnancements.

2.2.2 Spécification des problèmes d'ordonnancement

Les problèmes d'ordonnancement peuvent être référencés selon une ancienne forme de spécification [Conway et al., 1967]. Cependant, ils sont aujourd'hui le plus souvent exprimés selon la notation dite de Graham décomposée en trois champs successifs respectivement notés α , β et γ [Graham et al., 1979] [Blazewicz et al., 1996].

Le champ α décrit la structure du problème traité et l'organisation des ressources. Le champ β exprime les caractéristiques de tâches et l'ensemble des conditions d'exécution et contraintes explicites qui s'appliquent au système. Le champ γ indique la liste des objectifs visés représentant les critères que l'ordonnement cherche à optimiser.

En raison d'une grande diversité des problèmes d'ordonnement, il existe un grand nombre de valeurs possibles pour chacun des trois champs de la notation [Blazewicz, 1987] [Pinedo, 1995] [Lee et al., 1997] [Vignier, 1997] [T'kindt and Billaut, 2002].

Le champ α est initialement constitué de deux sous-champs notés α_1 et α_2 respectivement destinés à l'indication de la classe du système considéré et du nombre de ressources disponibles. Le tableau 2.2 énumère les valeurs du champ α les plus fréquentes.

Sous-champ α_1		Sous-champ α_2	
Valeur	Description	Valeur	Description
1	Machine unique	\emptyset	Nombre quelconque de machines ou d'étages
P	Machines parallèles identiques	m	Nombre défini de machines ou d'étages
Q	Machines parallèles proportionnelles		
R	Machines parallèles quelconques		
O	Open-shop		
J	Job-shop		
F	Flow-shop		
X	Mixed-shop		
OG	Open-shop généralisé		
JG	Job-shop généralisé		
FH	Flow-shop hybride		

TAB. 2.2: Valeurs usuelles du champ α

Des extensions ont été apportées au champ α sous la forme d'un ajout des sous-champs α_3 et α_4 pour exprimer les problèmes d'ordonnement spécifiques aux flow-shop hybrides [Vignier et al., 1999]. La spécification du champ α d'un système flow-shop hybride est mentionnée par $FHm, (\alpha_3\alpha_4^{(k)})_{k=1}^m$. L'expression de la structure d'un procédé constitué de trois étages intégrant respectivement une, trois et deux machines, tel que celui présenté par la figure 2.3 et en supposant que les machines parallèles d'un même étage sont toutes identiques, s'écrit $FH3, (1^{(1)}, P3^{(2)}, P2^{(3)})$. Il peut être simplifié et ramené à $FH3, (1, P3, P2)$ en considérant que les indications précisant la nature des

étages sont notées en respectant leur ordre d'implémentation sur le procédé modélisé. Le tableau 2.3 résume les valeurs d'affectation possibles pour les sous-champs α_3 et α_4 .

Sous-champ α_3		Sous-champ α_4	
Valeur	Description	Valeur	Description
1	Machine unique dans le pool	\emptyset	Nombre quelconque de machines à l'étage
P	Machines parallèles identiques	k	Nombre défini de machines à l'étage
Q	Machines parallèles proportionnelles		
R	Machines parallèles quelconques		

TAB. 2.3: *Extensions du champ α pour la spécification de flow-shop hybrides*

D'autres formalismes ont également été proposés pour permettre la spécification de systèmes mettant en œuvre une ressource partagée telle qu'un serveur destiné aux opérations de chargement et de configuration des machines préalablement au traitement des pièces [Hall et al., 2000] ou un robot chargé des opérations d'acheminement des pièces entre les machines [Hurink and Knust, 2001]. Les deux sous-champs α_1 et α_2 sont alors complétés par les indications $S1$, pour signifier l'implémentation d'un serveur, ou $R1$ dans le cas d'un robot. Le champ α est ainsi renseigné par $\alpha_1\alpha_2, S1$ ou $\alpha_1\alpha_2, R1$.

Le champ β est celui qui porte le plus grand nombre d'informations. Il fait constamment l'objet de nombreuses extensions. Initialement proposé avec six sous-champs, le champ β est aujourd'hui généralement présenté avec un ensemble de huit sous-champs, désignés par β_1 à β_8 . Le paramètre β_1 marque la possibilité de préemption des opérations dont l'exécution peut ainsi être morcelée. Le paramètre β_2 indique la présence de ressources auxiliaires. Le paramètre β_3 concernent les différentes relations de précedence liant les opérations entre elles. Le paramètre β_4 exprime le fait que la date de disponibilité est individualisée pour chaque opération. Les paramètres β_5 et β_6 décrivent respectivement les caractéristiques de l'échéance des opérations et leur durée de réalisation. Le paramètre β_7 exprime les contraintes temporelles portant sur l'enchaînement des opérations. Enfin, le paramètre β_8 indique, dans le cas de machines parallèles, des restrictions sur la polyvalence des machines. Lorsque le champ α est étendu par les valeurs $S1$ ou $R1$, des contraintes supplémentaires portant sur les temps d'exécution des opérations assignées à la ressource partagée peuvent être indiquées. Le tableau 2.4 présente un recensement des principales valeurs utilisées pour renseigner le champ β .

Valeur	Description	Valeur	Description
<i>prec</i>	Contraintes de précedence quelconques entre opérations	<i>tree</i>	Contraintes de précedence arborescentes entre opérations
<i>in - tree</i>	Contraintes de précedence arborescentes entrantes entre opérations	<i>out - tree</i>	Contraintes de précedence arborescentes sortantes entre opérations
<i>chains</i>	Contraintes de précedence chaînées entre opérations	<i>blcg</i>	Fin de traitement simultanée des machines
<i>sp - graph</i>	Contraintes de précedence sous forme de graphe série-parallèle entre opérations	<i>split</i>	Découpage des opérations en lots pour une exécution simultanée sur plusieurs machines
<i>no - wait</i>	Pas d'attente entre deux opérations successives d'un travail	<i>unavail_j</i>	Périodes connues d'indisponibilité des machines
<i>permu</i>	Permutation (uniquement pour le flow-shop)	<i>pmtn</i>	Préemption
<i>nmit</i>	Pas d'inactivité d'une machine lorsqu'elle peut exécuter une opération	<i>block</i>	Encours limités
<i>no - idle</i>	Pas de temps d'inactivité des machines	<i>recrc</i>	Plusieurs passages d'un travail sur une même machine
<i>res</i>	Ressources annexes	<i>over</i>	Recouvrement d'opérations successives d'un travail
\tilde{d}_j	Date impérative de fin de réalisation pour chaque travail	<i>res1</i>	Une ressource annexe
$p_j = p$	Temps de réalisation commun à tous les travaux	r_j	Date de disponibilité propre à chaque travail
$p_{ij} = p$	Temps de d'exécution des opérations commun à toutes les machines	p_j	Temps de réalisation arbitraire des travaux
$\underline{p} \leq p_{ij} \leq \bar{p}$	Temps d'exécution des opérations compris entre deux bornes	p_{ij}	Temps d'exécution des opérations arbitraire pour chaque machine
$d_j = d$	Date souhaitée de fin de réalisation commune à tous les travaux	$p_{ij} = 1$	Temps d'exécution des opérations unitaire pour toutes les machines
		d_j	Date souhaitée de fin de réalisation arbitraire pour chaque travail

TAB. 2.4: Valeurs usuelles du champ β

Le champ γ est renseigné par un ou plusieurs critères d'optimisation selon que l'ordonnement recherché est mono-objectif ou multi-objectifs. Les valeurs classiques sont donc celles présentées dans le tableau 2.1.

L'association des trois champs α , β et γ pour exprimer un problème d'ordonnement s'effectue sous la forme $\alpha/\beta/\gamma$. Lorsqu'ils sont plusieurs, les sous-champs des champs β et γ sont séparés par des virgules. Par exemple, le problème dont la formulation est $P/prec, r_j, \tilde{d}_j/C_{max}$ exprime un ordonnancement où les opérations sont soumises à une contrainte de précedence et doivent être exécutées dans une fenêtre de temps sur un ensemble de machines parallèles identiques. Le critère visé est la minimisation du makespan.

2.2.3 Méthodes de résolution

Les problèmes d'ordonnement présentent différents degrés de complexité. Différentes classes de méthodes de résolution ont alors été proposées en fonction du niveau de difficulté des problèmes [Mac Carthy and Liu, 1993] [Esswein, 2003]. Un annexe rappelle les notions de base de la complexité des problèmes et de leurs méthodes de résolution.

Les problèmes de classe \mathcal{P} sont résolus par des algorithmes polynomiaux garantissant en général l'optimabilité de la solution fournie. Lorsque les problèmes sont \mathcal{NP} -complets ou \mathcal{NP} -difficiles, la mise en œuvre d'algorithmes polynomiaux semble peu réaliste et il est alors souvent fait appel à des méthodes de résolution approchée basées sur des techniques *heuristiques* qui se contentent de produire une solution recevable sans pour autant garantir qu'elle soit la meilleure. Toutefois, des méthodes exactes peuvent parfois être appliquées pour traiter des problèmes de taille raisonnable. Elles concernent essentiellement la *programmation dynamique* et la méthode de *séparation et évaluation*.

Hormis les méthodes exactes et heuristiques, il est également possible d'ordonner des machines en appliquant des *règles de tri* basées sur des priorités. Contrairement aux autres méthodes de résolution précédemment citées, les règles de tri ne sont pas utilisées pour prévoir un ordonnancement global pour l'ensemble du système. Elles se limitent à la sélection d'un travail en vue de son traitement à chaque fois que la machine concernée devient disponible. Les décisions sont ainsi prises unitairement pour chaque machine tout au long de l'exécution du système. Cette méthode présente l'avantage de

pouvoir prendre en compte les perturbations et les aléas rencontrés durant la production.

Lorsque les choix sont définis avant le début d'exécution du système, les termes d'*ordonnancement hors ligne* ou *ordonnancement prévisionnel* sont employés pour qualifier le contexte de résolution. Il s'agit dans ce cas d'établir un plan de production préalablement au démarrage de l'activité des machines. L'utilisation de règles de tri consiste en un *ordonnancement en ligne*. Les choix sont alors effectués pendant l'exécution du système en fonction de sa situation.

2.2.3.1 Méthodes exactes

La programmation dynamique a été établie en 1957 [Bellman, 1957]. Son principe est fondé sur une approche récursive et repose sur une décomposition d'un problème de taille n en n problèmes de dimension 1. L'algorithme de résolution se déroule séquentiellement en n étapes et le passage d'une étape à une autre s'effectue à partir de lois d'évolution et d'une décision [Chevalier, 1977].

La méthode de séparation et évolution, couramment appelée *branch and bound*, a été proposée au début des années 60 [Land and Doig, 1960]. Elle s'appuie sur une énumération et une séparation de l'espace des solutions réalisables afin d'aboutir à un regroupement en plusieurs branches qui peuvent ensuite être évaluées par rapport à une borne inférieure visée. Les branches ne présentant pas d'évaluation inférieure à la meilleure solution connue sont progressivement éliminées.

2.2.3.2 Méthodes heuristiques

Les techniques à base d'heuristiques sont classées en deux grandes catégories que sont les *heuristiques constructives* et les *heuristiques amélioratrices*. Les heuristiques constructives sont des méthodes itératives dont le but est d'établir une solution à partir de données initiales en complétant une solution partielle à chaque itération. Dans le cadre d'un problème d'ordonnancement, il peut s'agir de placer successivement les opérations d'un travail ou d'enrichir à chaque itération la liste des opérations affectées à une ressource. Les heuristiques amélioratrices ne reposent pas sur une construction progressive de la solution. Elles s'attachent à modifier par étapes successives un résultat dans le but de l'améliorer progressivement jusqu'à obtenir la satisfaction d'un critère visé. La plupart des heuristiques amélioratrices utilise la notion de *voisinage* qui consiste

à affiner un résultat en partant d'une solution initiale pour ensuite calculer les solutions proches et en retenir une meilleure. Les méthodes de voisinage les plus courantes sont la *méthode de descente*, le *recuit simulé*, la *recherche tabou* et les *algorithmes génétiques*.

La méthode de descente est caractérisée par le fait qu'elle ne sélectionne, à chaque itération, que des solutions apportant une amélioration du résultat à partir d'une solution initiale donnée. Dans la plupart des cas, elle ne conduit pas à un résultat global optimal mais uniquement à un solution locale optimale dont la valeur est liée à la solution initiale adoptée. Son efficacité peut être améliorée par des applications multiples en prenant en compte des solutions initiales distinctes et en retenant le meilleur des résultats locaux obtenus.

La méthode du recuit simulé s'inspire des techniques thermodynamiques de traitement des métaux. Elle a été introduite en 1983 [Kirkpatrick et al., 1983] et se déroule de manière analogue à un processus alterné de cycles de refroidissement lent et de réchauffage destinés à minimiser l'énergie d'un matériau. Le principe est de modifier une solution initiale donnée afin de produire une seconde solution qui est conservée si elle améliore le critère visé. L'acceptation d'une plus mauvaise solution permet l'exploration d'une plus grande partie de l'espace de solutions et évite ainsi une limitation des recherches à un optimum local.

Le mécanisme de recherche tabou a été initié et s'est développé au début des années 80 [Glover, 1977] [Glover, 1986] et des variantes ont été proposées quelques années plus tard [Glover, 1989] [Glover, 1990]. Son principe consiste à examiner le voisinage de la solution courante à chaque itération et à retenir la solution voisine qui minimise la fonction objectif. Cependant, l'opération de sélection peut conduire à une augmentation du résultat si toutes les solutions du voisinage présentent une valeur supérieure à la solution courante. Ce mécanisme permet de sortir d'un optimum local et de poursuivre la recherche dans un ensemble plus étendu de solutions. Toutefois, afin d'éviter un phénomène de bouclage permanent entre deux solutions voisines, la recherche tabou utilise un effet de mémoire qui lui interdit de retenir les dernières solutions explorées. La méthode utilise ainsi une liste contenant les dernières solutions visitées.

Les algorithmes génétiques sont basés sur une imitation du phénomène d'adaptation des êtres vivants. Ils sont issus du travaux de recherche menés dès le début des années 60 [Holland, 1962] et qui ont véritablement abouti une quinzaine d'années plus tard [Holland, 1975]. Leur application aux problèmes d'optimisation a été formulée

en 1989 [Goldberg, 1989]. Les algorithmes génétiques, parfois appelés *algorithmes évolutionnaires*, s'appuient sur des mécanismes inspirés de la génétique et de l'évolution naturelle tels que le croisement, la mutation et la sélection. Le problème à résoudre est alors identifié à un environnement donné et les solutions correspondent à des individus évoluant dans cet environnement. À chaque génération, seuls les individus les mieux adaptés à l'environnement sont retenus. Ainsi, au bout d'un certain nombre de générations, il ne persiste que les individus les plus particulièrement adaptés à l'environnement et donc les plus aptes à répondre à la problématique.

2.2.3.3 Règles de tri

Les règles de tri, parfois appelées *règles de priorité* ou encore *règles de sélection*, sont des méthodes utilisées pour sélectionner le prochain travail parmi un ensemble de travaux en situation d'attente de traitement sur une machine M_i . La complexité algorithmique d'une règle de tri est alors limitée en $O(n)$.

Parce qu'elles se limitent au choix d'un travail parmi un ensemble de travaux en attente, les règles de tri sont parfois considérées comme répondant à des problèmes de sélection et non pas d'ordonnement [Hicks and Pongcharoen, 2006]. Ces derniers étant liés à un contexte de planification. Néanmoins, nous soulignons que, notamment dans le domaine des systèmes temps réel, les règles de tri sont souvent considérées comme étant des algorithmes d'ordonnement.

Bien que plus d'une centaine de règles aient été recensées dans la littérature du domaine [Panwalkar and Iskander, 1977], nous ne présentons ci-dessous que les plus courantes. Plusieurs critères permettant d'établir une forme de classification des règles de tri ont alors été proposés [Blackstone et al., 1982] [Holthaus and Rajendran, 1997] [Rajendran and Holthaus, 1999]. Une manière simple consiste à les distinguer selon la donnée évaluée pour établir le niveau de priorité d'un travail J_j sur une machine M_i :

- les règles basées sur la durée opératoire p_{ij} ,
- les règles basées sur la date d'échéance d_j ,
- les règles basées ni sur la durée opératoire ni sur la date d'échéance.

Les règles *SPT*, pour *shortest processing time*, et *LPT*, pour *longest processing time*, appartiennent à la première classe. Elles consistent à enchaîner les travaux sur la machine M_i respectivement dans l'ordre croissant et décroissant de leur durée opéra-

toire p_{ij} . Elles sont plus particulièrement utilisées pour résoudre des problèmes d'ordonnement dans les systèmes de type flow-shop. La règle *SPT* est optimale pour la résolution de problèmes cherchant à optimiser la somme des dates d'achèvement de travaux C_{sum} [Blazewicz et al., 1996]. Les règles *SPT* et *LPT* se rencontrent également sous la forme d'une variante où les travaux sont affectés d'une pondération w_j . Il s'agit des règles *WSPT* et *WLPT* pour *weighted shortest processing time* et *weighted longest processing time* [Smith, 1956]. La règle *WSPT* est aussi très souvent désignée comme étant la règle de Smith.

La règle de Jackson, ou *EDD* pour *earliest due date*, dont le principe repose sur un ordonnancement des travaux selon l'ordre croissant de leur date d'achèvement souhaitée d_j , est optimale dans le cas où le critère visé concerne la minimisation du retard algébrique maximal L_{max} ou le retard absolu maximal T_{max} [Jackson, 1955]. L'algorithme de Moore [Moore, 1968], parfois appelé algorithme de Moore-Hodgson, est basé sur la règle de Jackson. Il est optimal dans le cas où la fonction objectif vise la minimisation des travaux en retard. La règle dite du *ratio critique* et notée *CR* s'appuie sur un calcul du rapport du temps restant avant la date de fin de réalisation souhaitée d'un travail $d_j - t$ à la durée opératoire totale restante à l'instant t . Les travaux sont traités dans l'ordre décroissant de leur ratio critique. La règle *LST*, pour *least slack time*, évalue l'urgence d'un travail en fonction de la marge temporelle disponible avant la date d'achèvement souhaitée. La plus haute priorité est alors affectée au travail dont le délai avant le début de son traitement au plus tard est le plus court. Une variante appelée *LST/OPN*, pour *least slack time per operation*, est considérée comme l'une des règles de tri les plus performantes parmi celles basées sur la date d'échéance des travaux. Son principe repose sur la sélection du travail qui présente la plus petite marge par rapport au nombre d'opérations restant à exécuter et, s'il est admis que chaque machine exécute une opération, au nombre de machines restant à visiter avant son achèvement complet sur la dernière machine k . Ces quatre règles de tri sont classées dans la seconde classe.

En dehors, des règles basées sur la durée opératoire ou la date d'échéance des travaux évoquées ci-dessus, nous pouvons citer les principes de sélection les plus classiques que sont *FIFO*, pour *first in first out*, et *LIFO*, pour *last in first out*. Elles s'attachent à traiter les travaux respectivement dans l'ordre croissant et décroissant de leur date de disponibilité r_{ij} . La règle *FIFO* sert souvent de référence à des fins de comparaison avec les autres règles de tri [Rose, 2003] [Chong et al., 2005] [Christopher et al., 2005].

La règle *AT*, pour *arrival time* est une variante de *FIFO* qui impose la sélection du travail dont la date d'entrée d_j sur le système est la plus ancienne. La règle de tri aléatoire, notée *RND*, doit aussi est mentionnée. Ces quatre règles de tri s'inscrivent dans la troisième classe.

Des combinaisons de règles de tri ont été étudiées dans le but de rechercher une amélioration de l'efficacité dans la résolution de problèmes spécifiques [Grabot et al., 1993] [Holthaus and Rajendran, 1997]. Les cas d'utilisation de règles combinées sont parfois présentés comme une quatrième classe.

Le tableau 2.5 rappelle les principales règles de tri utilisées pour évaluer chaque travail j d'un ensemble de travaux en attente de traitement sur une machine M_i et sélectionner celui qui présente la plus haute priorité.

Règle	Calcul	Règle	Calcul
<i>FIFO</i>	$\min_{J_j \in \mathcal{J}} \{r_{ij}\}$	<i>LIFO</i>	$\max_{J_j \in \mathcal{J}} \{r_{ij}\}$
<i>RND</i>	Tirage aléatoire	<i>EDD</i>	$\min_{J_j \in \mathcal{J}} \{d_j\}$
<i>SPT</i>	$\min_{J_j \in \mathcal{J}} \{p_{ij}\}$	<i>LPT</i>	$\max_{J_j \in \mathcal{J}} \{p_{ij}\}$
<i>AT</i>	$\min_{J_j \in \mathcal{J}} \{r_j\}$	<i>LST</i>	$\min_{J_j \in \mathcal{J}} \left\{ d_j - t - \sum_{m=i}^k p_{mj} \right\}$
<i>CR</i>	$\min_{J_j \in \mathcal{J}} \left\{ \frac{d_j - t}{\sum_{m=i}^k p_{mj}} \right\}$	<i>LST/OPN</i>	$\min_{J_j \in \mathcal{J}} \left\{ \frac{d_j - t - \sum_{m=i}^k p_{mj}}{k - i - 1} \right\}$

TAB. 2.5: Principales règles de tri

Le choix de la règle de tri la plus adaptée pour répondre à un problème d'ordonnancement donné est le plus souvent un travail difficile à mener. Des études comparatives [Haupt, 1989] [Rajendran and Holthaus, 1999] ainsi que des travaux de thèse portant spécifiquement sur le sujet [Boucon, 1991] ne permettent pas véritablement d'aboutir à des résultats applicables quelque soit le problème posé. De manière générale, il est communément admis que les règles de tri basées sur la durée opératoire des travaux sont plus les efficaces dans des conditions de charge importante, tandis que les règles basées sur la date d'échéance aboutissent à de meilleurs résultats dans le cas

contraire [Reddi and Sadowsky, 1976] [Blackstone et al., 1982].

Lorsqu'une règle de tri permet de répondre à un problème d'ordonnancement donné, il est admis qu'elle est *optimale* si la solution à laquelle elle aboutit est au moins aussi performante que celle obtenue avec n'importe quelle autre règle de tri.

2.2.4 Évaluation des solutions

Un ordonnancement donné est une solution potentielle à un problème d'ordonnancement. Les ordonnancements sont précisément définis par les dates de début et de fin de traitement de chaque tâche. Dans le cas d'un problème d'atelier, l'activité du système de production est donnée par les dates de début et de fin d'exécution de chaque opération de l'ensemble des travaux à réaliser par les machines. Les ordonnancements sont alors généralement répartis en quatre classes en fonction du contexte de succession des exécutions d'opérations sur les machines.

Un ordonnancement est *admissible* ou *faisable* lorsque la solution qu'il représente respecte toutes les contraintes du problème. Il est considéré comme *actif* quand il ne permet pas d'avancer l'exécution d'une opération sans provoquer le retard d'une autre. Il est dit *semi-actif* si aucune exécution d'opération ne peut être avancée sans changer l'ordre d'exécution des autres opérations sur les machines. Enfin, un ordonnancement est appelé *sans retard* quand aucune machine n'est laissée inactive lorsqu'une opération qui lui est affectée est prête pour son exécution [Dupas, 2004]. Un ordonnancement actif est donc toujours semi-actif et un ordonnancement sans délai est forcément actif.

2.3 Conclusion

Ce chapitre décrit le cadre général de la théorie de l'ordonnancement et son application aux problèmes d'atelier. Nous présentons une typologie des systèmes de production ainsi que les contraintes et conditions d'exécution auxquelles ils sont soumis. Le contexte dans lequel les techniques d'ordonnancement sont utilisées pour servir au mieux la performance des procédés manufacturiers est également exposé.

Les problèmes d'ordonnancement d'atelier sont le plus souvent \mathcal{NP} -complets ou \mathcal{NP} -difficiles et leur résolution repose alors sur l'utilisation de méthodes approchées. Toutefois, dans certains cas, ils sont de complexité moindre et peuvent être résolus à

l'aide de méthodes exactes. Il est également largement fait usage de règles de tri pour répondre à des problèmes d'ordonnancement dans un contexte de production industrielle. Dans ce cas, les choix sont effectués durant l'exécution du système.

Ce chapitre constitue une synthèse des problèmes d'ordonnancement rencontrés dans la mise en œuvre de systèmes de production. Il doit ainsi permettre au lecteur de situer le contexte dans lequel se situent les travaux de cette thèse.

Chapitre 3

Description des lignes d'assemblage considérées

Sommaire

3.1	Généralités sur les lignes d'assemblage	48
3.1.1	Historique	49
3.1.2	Modèles théoriques et problèmes d'ordonnancement classiques	49
3.2	Caractéristiques de base des lignes d'assemblage étudiées	53
3.2.1	Considérations principales	53
3.2.2	Données statiques	54
3.2.3	Données dynamiques	57
3.2.4	Évaluation de la cadence de production	57
3.2.5	Calcul du makespan	62
3.2.6	Latence et périodicité d'activité des machines	64
3.2.7	Exemples de systèmes simulés	65
3.2.7.1	Cadence fixée par la machine la plus lente	66
3.2.7.2	Cadence fixée par une phase de transfert	68
3.3	Mise en œuvre d'un contrôleur central	70
3.3.1	Notions de contrôle et de commande	70
3.3.2	Structure du système de pilotage	71
3.3.3	Charge du contrôleur et interactions avec les machines	72
3.4	Expression de la problématique	75
3.4.1	Contrainte sur le temps de réponse du contrôleur	75
3.4.2	Spécification du problème d'ordonnancement	76
3.5	Conclusion	77

Ce chapitre a pour objet de définir le cadre général des travaux en décrivant les procédés flow-shop étudiés et en précisant les spécificités qui caractérisent le modèle que nous considérons. Les notations que nous retenons dans le cadre de cette thèse sont également introduites.

Nous apportons en premier lieu quelques informations générales relatives à l'histoire de la mise en œuvre de lignes d'assemblage dans le monde industriel. Nous présentons ensuite les principaux modèles théoriques et les problèmes d'ordonnement les plus couramment étudiés dans la littérature.

Dans la seconde section, les paramètres des systèmes d'assemblage considérés sont détaillés puis nous apportons les éléments qui permettent de déterminer a priori la cadence maximale de production et d'évaluer le makespan. Nous présentons également deux exemples numériques provenant de la simulation de systèmes dont les résultats sont exprimés sous la forme de diagrammes de Gantt.

La troisième section décrit la particularité des systèmes flow-shop que nous étudions. Elle résulte de l'utilisation d'un contrôleur central interagissant avec l'unité de commande des différentes machines de production. Le contrôleur est ainsi chargé d'assurer le conditionnement et la coordination de l'activité des machines. La fonction de pilotage est alors hiérarchisée et déclinée sur deux niveaux. Le contrôleur, implanté à l'étage supérieur, représente une ressource partagée par l'ensemble des machines de production situées à l'étage inférieur.

La problématique induite par l'implantation d'un contrôleur central et la hiérarchisation de la fonction de pilotage est ensuite exposée. Nous présentons alors l'objectif visé par ce travail de thèse. Enfin, deux extensions à la notation de Graham sont proposées afin de spécifier le problème d'ordonnement que nous considérons. Elles sont destinées à permettre l'expression de la particularité de l'organisation des ressources et à signifier une restriction quant à l'application du mécanisme de préemption.

3.1 Généralités sur les lignes d'assemblage

Les lignes d'assemblage sont des équipements du type flow-shop. Elles sont mises en œuvre pour aboutir à des réductions de coûts dans le cas de fabrication de masse de produits standardisés et sont surtout destinées à l'exécution de tâches répétitives et très spécialisées [Shtub and Dar-El, 1989].

L'industrie manufacturière fait aujourd'hui largement appel à ce type de procédés pour répondre aux objectifs de productivité imposés par le contexte économique dans lequel elles évoluent.

3.1.1 Historique

Le concept des lignes d'assemblage trouve son origine dans la théorie de l'organisation scientifique du travail proposée par Frederic Winslow TAYLOR (1856-1915) dans son ouvrage « The principles of Scientific Management » paru en 1911. Leur principe est basé sur la division du travail par son fractionnement en tâches simples et répétitives.

La généralisation des lignes d'assemblage est directement liée à la croissance de l'industrie automobile aux États-Unis et en Europe au début du XX^e siècle. En 1913, Henry FORD (1863-1947) et ses ingénieurs ont été parmi les premiers à concevoir des usines entières destinées à accueillir des lignes d'assemblage pour la production du fameux modèle T. Les véhicules en cours d'assemblage étaient ainsi acheminés par des convoyeurs automatisés vers les différents postes de travail occupés par des opérateurs. Les coûts de production étaient réduits par l'accélération du temps d'assemblage ramené de 28 heures à 93 minutes [Yates, 1983].

L'activité sur les différents postes de travail était alors surtout manuelle. De nos jours, les équipements sont de plus en plus automatisés et le nombre d'opérateurs est généralement faible.

3.1.2 Modèles théoriques et problèmes d'ordonnancement classiques

Les systèmes de production en ligne ont été déclinés en une multitude de modèles théoriques [Färber and Coves, 2005] [Boysen et al., 2006]. Ils diffèrent généralement par les contraintes imposées lors de leur exécution, l'organisation des machines ou de la diversité des modèles de pièces fabriquées. Aussi, de nombreuses extensions sont couramment spécifiées pour enrichir le modèle de base afin de mieux répondre à l'analyse de cas pratiques ou industriels [Espinouse, 1998] [Finke et al., 2002].

Les problèmes d'ordonnancement sous-jacents couvrent un large éventail de recherches et un grand nombre d'ouvrages et de publications ont été consacrés à ce domaine depuis

plusieurs décennies [Gupta and Stafford, 2006]. Ils s'attachent le plus souvent à optimiser le makespan par la minimisation de la date de fin de fabrication de la dernière pièce [Palmer, 1965] [Campbell et al., 1970] [Chu et al., 1995]. Le problème du flow-shop consiste classiquement à établir un ordre de passage des pièces sur les machines. Pour un lot de fabrication constitué de n pièces ayant chacune une durée de rétention propre sur chacune des k machines, le nombre maximal de séquences différentes à évaluer pour établir l'ordre de lancement de la fabrication est de $(n!)^k$. L'objectif est alors orienté vers la recherche d'un ordonnancement qui minimise la durée de réalisation d'un ensemble donné de travaux ou tout autre critère.

Le problème d'ordonnancement classique lié à un système flow-shop constitué de deux machines sans moyen de transport avec pour critère la minimisation du makespan, exprimé selon la notation de Graham par $F2//C_{max}$, représente la base des travaux de recherche portant sur les systèmes à cheminement unique. Il a été résolu à l'aide d'un algorithme de complexité $O(n \log(n))$ au cours des années 50 [Johnson, 1954].

Le problème d'ordonnancement des systèmes flow-shop est prouvé comme étant de type \mathcal{NP} -complet lorsque le modèle étudié est basé sur un nombre plus élevé de machines [Garey et al., 1976] ou que le critère visé est la somme des dates d'achèvement des travaux [Garey and Johnson, 1979]. Il l'est également quand la fonction objectif s'attache à minimiser le retard maximal algébrique [Lenstra et al., 1977]. De nombreux algorithmes utilisant des méthodes approchées pour résoudre les différents cas généraux ou pour minimiser le temps total de séjour des pièces sur le système sont disponibles dans la littérature [Kohler, 1975] [Ahmadi and Bagchi, 1990] [Rajendran and Chandhuri, 1991] [Davis and Kenet, 1993]. Il faut cependant relever qu'une solution exacte basée sur une méthode de séparation et évaluation a également été proposée pour traiter le cas spécifique du flow-shop classique constitué de trois machines [Houari and Daouas, 1999].

De manière générale, la minimisation du makespan est majoritairement abordée par les travaux portant sur les systèmes flow-shop [Gupta, 1972] [Baker, 1975] [Taillard, 1990] [Widmer and Hertz, 1989]. Les études concernant l'optimisation des dates d'achèvement de travaux sont également courantes [Bansal, 1977] [Szwarc, 1983].

Dans le cas où les opérations peuvent être interrompues, le problème d'ordonnancement du flow-shop à deux machines se résout polynomialement si l'objectif concerne la minimisation du makespan. Cependant, un algorithme polynomial ne suffit plus à sa résolution lorsque le nombre de ressources augmente et que le critère visé concerne la

somme des dates d'achèvement des travaux [Du and Leung, 1993] ou le retard algébrique maximal [Gonzalez and Sahni, 1978] [Cho and Sahni, 1981].

Lorsque le problème d'ordonnancement d'un système flow-shop est abordé dans un contexte industriel, les procédés sont souvent étudiés en tenant compte d'une impossibilité de dépassement des pièces en cours de fabrication. La contrainte de permutation est alors appliquée au modèle [Potts et al., 1991] [Carlier and Rebaÿ, 1996] [Rebaine, 2005]. L'ordre de passage sur les machines est alors unique et le nombre de séquences à évaluer est considérablement réduit à $n!$. Nous pouvons ajouter que la situation est identique lorsque toutes les pièces sont fabriquées selon une gamme opératoire unique.

Les contraintes d'exécution sans temps d'arrêt des machines ou sans phase d'attente des pièces sont fréquemment prises en compte dans l'étude des systèmes flow-shop [Saadani, 2003] [Kamburowski, 2004] [Saadani et al., 2005] [Wang et al., 2005].

Les travaux basés sur une contrainte de limitation des capacités d'encours ont également fait l'objet de nombreuses publications depuis le début des années 90 [Leisten, 1990] [Brucker et al., 2003] [Soukhal et al., 2005] [Wang et al., 2006].

Pour un flow-shop classique, la date de disponibilité des travaux est par défaut fixée à l'instant 0 de l'exécution du système. Les travaux sont ainsi tous considérés comme disponibles pour leur réalisation dès l'initialisation du système. Cependant, des travaux concernent des systèmes flow-shop pour lesquels la date de disponibilité des travaux est individualisée [Potts, 1985] [Kashyrskikh et al., 2001].

Les systèmes flow-shop avec serveur représentent une variante assez rarement étudiée dans la littérature [Iravani and Teo, 2005] [Brucker et al., 2005]. De tels procédés font appel à une ressource particulière dont l'activité est partagée entre les différentes machines du système pour assurer l'exécution spécifique de tâches de configuration. Le serveur peut être une machine ou un opérateur.

Une autre variante moins souvent abordée des systèmes flow-shop concerne les modèles intégrant un équipement central destiné au transfert des pièces entre les différentes machines [Hurink and Knust, 2001] [Kats and Levner, 2002]. Un tel modèle est qualifié de *flow-shop robotisé* et peut être considéré en tenant compte de fenêtres de temps finies ou infinies pour effectuer les transferts de pièces. Le cas désigné comme étant le *problème d'ordonnancement de la grue*, couramment désigné par *hoist scheduling problem*, est un exemple de flow-shop robotisé [Philips and Hunger, 1976] [Lim, 1997]

[Hindi and Fleszar, 2004]. Il consiste à ordonnancer simultanément l'exécution bornée des opérations de production et celles de transport.

Sur un plan général, le mode de transfert de pièces entre les machines constitue une caractéristique importante des systèmes de production organisés en ligne. Les phases de transfert sont le plus souvent automatisées pour satisfaire les exigences de performance attendues des systèmes de production actuels [Liu and Mac Carthy, 1996]. Les systèmes peuvent ainsi être caractérisés par la synchronisation ou la désynchronisation de leurs phases de transfert [Buzacott and Shanthikumar, 1993] [Anderson, 1997].

Lorsque les phases de transfert sont synchronisées, il faut distinguer le transfert déclenché par le terme du cycle opératoire de la machine la plus lente de celui basé sur un temps de cycle global et ainsi activé à intervalles réguliers. Ces types de systèmes sont usuellement désignés par le terme anglais *paced flow-shop*. L'implantation peut être linéaire ou circulaire, les machines sont alors disposées autour d'un plateau tournant [Groover, 1996].

Les systèmes flow-shop intégrant un transfert désynchronisé autorisent à une machine ayant achevé son cycle opératoire de relâcher la pièce détenue pour permettre son acheminement vers la machine suivante. Ce type, habituellement appelé *unpaced flow-shop*, impose la mise à disposition d'espaces tampon entre les étages du système flow-shop afin de stocker les pièces en attente de leur traitement par la machine suivante. Les espaces tampon permettent ainsi d'absorber le découplage opératoire induit par les différences de temps de cycle opératoire des machines.

Nous citons également le cas des lignes d'assemblage au sein desquelles les temps de cycle opératoire sont propres à chaque machine dont la durée de traitement d'une pièce est ainsi indépendante de celle des autres ressources du système. Les problèmes relevant d'une telle spécificité structurelle sont souvent désignés par le terme anglais *machine correlation problems* [Watson et al., 1999] [Watson et al., 2002]. L'activité autonome et le comportement asynchrone des machines qui en découlent représentent l'une des principales caractéristiques des systèmes d'assemblage flexibles [Donath and Graves, 1988] [Chetty and Gnanasekaran, 1996]. Ce type de problèmes spécifiques se différencie de deux autres classifications proposées dans la littérature et désignées par *job correlation problems* et *time gradients problems* [Rinnooy Kan, 1976]. Ces problèmes s'appliquent à des cas pour lesquels les temps d'exécution sont uniquement liés aux travaux ou sont distribués en fonction de l'ordre d'implantation des machines sur le système.

3.2 Caractéristiques de base des lignes d'assemblage étudiées

Nos travaux concernent des systèmes de production industriels organisés en ligne. Il s'agit de procédés d'assemblage automatisés déclinés en une série de machines de production successivement visitées dans l'ordre de leur implantation par les pièces à fabriquer.

Nous présentons dans cette section la description principale des systèmes étudiés en décrivant l'architecture et le comportement des automatismes assurant l'activité des machines et l'acheminement des pièces. Nous précisons également les données numériques qui permettent la spécification des modèles.

3.2.1 Considérations principales

Le système est constitué des machines de production et d'un équipement de convoyage assurant le transfert des pièces sur l'ensemble du procédé. Chaque machine est individuellement commandée par un contrôleur interne, typiquement un automate programmable industriel. Nous appelons *unité de commande* le contrôleur interne à chaque machine de production.

L'acheminement des pièces à produire d'une machine vers une autre est assuré par un système de transfert automatisé et les phases de transfert sont désynchronisées. L'ordre de passage des pièces sur chacune des machines respecte celui de leur date d'entrée dans le système. Aucune pièce ne peut donc dépasser celle qui la précède. Par conséquent, le modèle considéré est un flow-shop de permutation. De plus, nous considérons que la date de disponibilité d'un travail correspond à sa date d'entrée au plus tôt sur le système. Un travail n'entre sur le système flow-shop que lorsque la première machine vient de libérer la pièce associée au travail précédent. Seul le premier travail présente une date de disponibilité égale à l'instant 0. Nous avons donc $\forall J_j \in [J_2, J_n] : r_{j-1} < r_j$ avec $r_1 = 0$.

Conformément aux considérations classiques retenues lors de l'étude de systèmes de production, les machines sont des ressources disjonctives. Il n'existe donc aucune possibilité d'exécution concurrente d'opérations sur une même machine et nous relevons également que les opérations ne sont pas préemptibles. Chaque ressource du système ne peut donc retenir plus d'une pièce à chaque instant et se limite ainsi au traitement

d'un seul travail ininterrompu. Le mécanisme de préemption ne s'applique donc pas aux machines de production.

Nos travaux sont basés sur l'analyse d'un procédé industriel existant, la capacité d'encours entre deux machines est donc limitée. La saturation d'un tel espace tampon provoque la suspension d'activité de la machine située en amont au terme de l'exécution des opérations relatives à la pièce en cours de traitement. Cette pièce n'est alors relâchée que lorsque la machine implantée en aval prend en charge la première pièce en situation d'encours à son entrée.

Le système étudié est de type à modèles multiples. Toutefois, nous admettons que la durée de rétention des pièces sur une machine donnée est constante quelque soit le modèle en cours de fabrication. L'intervalle temporel d'étude se limite en effet à un lot de production de pièces identiques. Cette hypothèse simplificatrice est a priori sans conséquence sur l'analyse en raison de l'activation exclusive d'un seul modèle à chaque instant. Aussi, nous retenons que les traitements de travaux successifs sur une machine donnée sont tous d'une durée équivalente.

Les systèmes de production étudiés sont alors des outils de production linéaires dont l'activité est caractérisée par des enchaînements identiques d'opérations réparties et exécutées sur un ensemble de ressources dans le but de réaliser des travaux. Les pièces à produire visitent successivement les ressources et quittent le système dans l'ordre exact de leur entrée. La fonction objectif que nous visons concerne la minimisation du makespan. En tenant compte des caractéristiques de base des lignes d'assemblage considérés dans le cadre de nos travaux et en appliquant la notation de Graham, nous pouvons décrire le problème d'ordonnancement sous-jacent par l'expression $F/perm, block, r_j/C_{max}$.

3.2.2 Données statiques

Les données statiques du modèle d'un système représentent l'ensemble des informations numériques nécessaires à son paramétrage préalablement à son exécution. Nous utilisons les notations courantes que nous enrichissons de certains éléments spécifiques pour exprimer avec précision les problèmes auxquels nous nous intéressons.

Conformément aux notations classiquement retenues dans le cadre de l'expression de problèmes d'ordonnancement d'atelier, l'ensemble des machines de production, représentant les ressources disponibles sur le système, est noté $\mathcal{M} = \{M_1, M_2, \dots, M_k\}$. L'assem-

blage de pièces effectué par les machines correspond à la réalisation d'un ensemble de travaux désignés par $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$. Les valeurs k et n représentent alors le nombre de machines et de travaux à réaliser au cours de la production. Comme nous admettons par simplification qu'une production ne concerne qu'un seul modèle, tous les travaux sont réalisés en respectant une gamme de fabrication unique. La durée d'exécution d'une opération donnée est identique pour tous les travaux. Nous limitons donc l'expression de l'ensemble des opérations à une unique gamme opératoire. De plus, et bien qu'une machine exécute au moins une opération, les hypothèses d'étude sur lesquelles nous nous basons ne restreignent pas la charge d'activité assignée à une machine M_i à une seule opération mais à m_i opérations. L'ensemble des opérations affectées à une machine M_i est alors désigné par $\mathcal{O}_i = \{O_{i,1}, O_{i,2}, \dots, O_{i,m_i}\}$.

Le temps d'exécution cumulé du jeu d'opérations \mathcal{O}_i assigné à la machine M_i est noté θ_i . Il est aussi appelé *durée de traitement opératoire*, ou plus simplement *durée opératoire*, et correspond implicitement à la durée de présence minimale d'une pièce sur la machine M_i . L'indice du terme θ_i est complété par le numéro de chaque opération pour exprimer la durée d'exécution de l'opération associée. Ainsi, la notation $\theta_{i,3}$ exprime la durée nécessaire à l'exécution de la troisième opération affectée à la machine M_i et désignée par $O_{i,3}$. Les délais de convoyage des pièces entre deux machines consécutives M_{i-1} et M_i ne sont généralement pas considérés comme nuls. Ils sont désignés par $\psi_{i-1,i}$.

Par analogie avec la notation C_{max} utilisée pour désigner le makespan, nous exprimons la durée effective de réalisation totale d'un unique travail par C_{min} afin de marquer la durée d'exécution minimale du système correspondant à la production d'une seule pièce. Elle est de fait égale à la somme des durées de traitement opératoire sur chacune des machines et des durées de convoyage de la pièce associée :

$$C_{min} = \sum_{i=1}^k \theta_i + \sum_{i=2}^k \psi_{i-1,i} \quad (3.1)$$

La valeur C_{min} est donc un temps incompressible correspondant à la durée minimale de séjour d'une pièce sur le système flow-shop. Elle dénote d'une situation optimale qui n'est généralement atteinte que pour la réalisation du premier travail. Dans ce cas particulier, les machines sont toutes disponibles et la pièce associée n'observe aucune

phase d'attente.

La durée cumulée des opérations exécutées par une machine n'est pas nécessairement égale à celle des machines adjacentes et il n'existe pas de synchronisation du cycle opératoire des différentes machines. Le temps de rétention des pièces par les machines n'est alors pas homogène et un découplage opératoire est ainsi observé entre une machine M_{i-1} et sa suivante M_i . Il a pour conséquence de générer un encours de travaux en attente de traitement par la machine M_i lorsque cette dernière présente une durée de rétention des pièces plus importante que celle de la machine M_{i-1} , soit $\theta_{i-1} < \theta_i$. Les espaces inter-machines constituent dans ce cas des zones tampons dont la capacité est finie et connue. Elle se note $\varpi_{i-1,i}$ et désigne le nombre maximal de travaux en encours entre les machines successives M_{i-1} et M_i .

Afin de définir la notion d'encours, nous considérons que toute pièce relâchée par une machine M_{i-1} et non encore prise en charge par la machine suivante M_i est en situation d'encours. Ainsi, une pièce en phase de transfert constitue également un encours. L'encours représente donc l'ensemble des pièces se trouvant à un instant donné entre deux machines consécutives. En conséquence, la capacité d'encours minimale est fixée à une pièce, soit $\forall M_i \in [M_2, M_k] : \varpi_{i-1,i} \geq 1$.

La figure 3.1 présente le modèle de base du système étudié. Les temps opératoires et les délais de transfert de pièces sont indiqués ainsi que les capacités d'encours qui sont notées entre parenthèses.

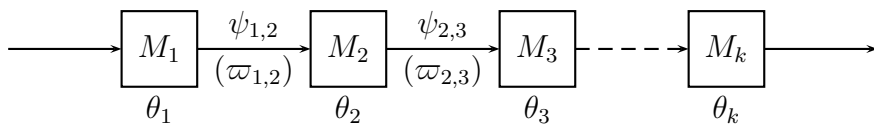


FIG. 3.1: *Modèle de base des systèmes étudiés*

Lorsque le nombre de travaux entre deux machines consécutives M_{i-1} et M_i atteint la capacité d'encours $\varpi_{i-1,i}$, la machine M_{i-1} suspend son activité au terme du traitement de la pièce qu'elle détient et retarde la prise en charge d'une éventuelle pièce en attente à son entrée. La pièce dont le traitement vient d'être achevé n'est pas relâchée. La reprise d'activité est conditionnée par le début de traitement par la machine M_i de la première pièce en encours.

De manière générale, nous retenons comme hypothèse dans la suite de cette thèse que

les systèmes étudiés évoluent de manière déterministe et que les durées opératoires θ_i et de transfert $\psi_{i-1,i}$ sont constantes d'un travail à un autre.

3.2.3 Données dynamiques

Les données dynamiques correspondent aux informations numériques recueillies au cours de l'exécution des systèmes étudiés.

Les dates de début et d'achèvement d'un travail J_j sur une machine M_i sont respectivement notées β_j^i et ξ_j^i . La durée totale de réalisation d'un travail J_j par le système modélisé correspond alors à $\xi_j^k - \beta_j^1$.

Les situations d'encours de travaux impliquent des phases d'attente et allongent le temps de présence des pièces sur le système. Ainsi, pour un travail j donné, la relation $\xi_j^k - \beta_j^1 \geq C_{min}$ est toujours respectée. L'égalité $\xi_j^k - \beta_j^1 = C_{min}$ dénote que la pièce associée au travail J_j n'a observé aucune phase d'attente au cours de sa production. Ce cas de figure se rencontre systématiquement pour la première pièce produite lorsque le système est en situation initiale et qu'il n'existe aucun travail préalable en cours. Toutefois, il est également constaté quand les machines sont implantées sur le système dans l'ordre décroissant de leur durée opératoire θ_i , soit $\forall M_i \in [M_2, M_k] : \theta_{i-1} > \theta_i$.

Puisque $\xi_j^k - \beta_j^1$ permet d'établir la durée de bout en bout de réalisation d'un travail J_j , alors le délai pour exécuter un ensemble fixé de n travaux par un système constitué de k machines, c'est-à-dire le makespan, peut être aisément déduit. Il coïncide avec la date de fin de réalisation du dernier travail :

$$C_{max} = \xi_n^k \quad (3.2)$$

Nous admettons en effet que la date de début de réalisation du premier travail sur la première ressource β_1^1 correspond à l'instant 0 de l'exécution du système.

3.2.4 Évaluation de la cadence de production

Plusieurs phases de fonctionnement peuvent être observées au cours de l'activité d'un système de production. Nous considérons que l'état initial correspond à une situation pour laquelle aucune pièce n'est présente sur le système. Les machines entrent alors successivement en production dans l'ordre de leur implantation sur le système jusqu'à ce

que leur activité se stabilise et respecte un cycle périodique. La montée en cadence d'un système de production correspond à un *régime transitoire*. Lorsque le comportement périodique de toutes les machines est atteint, le système fonctionne en *régime stationnaire* ou *régime permanent*. Le terme d'une production peut aussi s'effectuer selon un régime transitoire s'il est admis que le nombre de pièces entrant sur le système est égal au nombre de pièces à fabriquer. Le système est alors vidé lorsque la dernière pièce est achevée et libérée par la dernière machine.

En considérant le régime stationnaire d'un système flow-shop classique, un certain nombre de propriétés essentielles ont été étudiées et démontrées dans la littérature depuis plusieurs décennies [Johnson, 1954] [Cohen et al., 1983] [Iyama and Ito, 1987] [Terracol and David, 1987] [Commault and Dallery, 1990] [Gershwin, 1993]. Sur un plan général, ces propriétés permettent d'évaluer le niveau de performance des systèmes. La *cadence de production*, notée τ , est une caractéristique de performance importante qui peut être exprimée à partir du délai s'écoulant entre deux achèvements successifs de travaux. Elle correspond alors une valeur moyenne des durées constatées s'écoulant entre les dates de fin de fabrication de deux pièces consécutives au cours de la durée d'activité considérée du système de production :

$$\tau = \frac{1}{n} \sum_{i=1}^{n-1} (\xi_{i+1}^k - \xi_i^k) \quad (3.3)$$

La cadence de production est imposée par un élément du procédé contraignant le processus de production. Le terme *goulot d'étranglement* est utilisé pour évoquer un équipement dont la sollicitation est supérieure à la capacité et dont les effets sont de borner quantitativement la performance d'un système [Umble and Srikanth, 1990]. Il s'agit alors d'une ressource critique du procédé dont le délai de rétention des pièces fixe une borne sur la cadence de production, et donc sur le délai moyen d'inter-fins de travaux, pour l'ensemble du système [Hopp et al., 2005]. Le terme *ressource menante* est aussi parfois utilisé pour désigner le goulot d'étranglement d'un système de production.

La cadence de production d'un système flow-shop classique peut être conditionnée selon deux manières distinctes, chacune étant liée à une ressource particulière du procédé [Gershwin and Dallery, 1992] :

- la machine la plus lente,

– la phase de transfert la plus défavorable.

Dans le cas d'un système flow-shop classique, le délai moyen d'inter-fins de travaux est le plus souvent fixé par la machine dont la durée de déroulement du cycle opératoire est la plus importante du système. La cadence de production est alors donnée par le temps opératoire de la machine la plus lente :

$$\tau = \max_{M_i \in \mathcal{M}} \{\theta_i\} \quad (3.4)$$

Ce calcul tient compte du fait que la durée opératoire d'une machine donnée est considérée constante pour la réalisation de tous les travaux. Au cours de la réalisation de n travaux, les machines exécutent en effet toujours le même cycle opératoire durant la même durée.

La cadence de production peut également être fixée par la phase de transfert et la capacité de stockage de pièces entre deux machines successives M_{i-1} et M_i . Le délai moyen d'inter-fins de travaux est alors supérieur au temps opératoire de la machine la plus lente. Dans ce cas, il est alors constaté que les dates de sortie de pièces ne respectent pas une période stricte et les fins de travaux s'effectuent selon des rafales de pièces. Cependant, cette situation ne se rencontre en principe pas dans le cas de lignes d'assemblage automatisées pour lesquelles le temps de transfert est généralement réduit au maximum. Lorsqu'elle est observée, elle relève plutôt d'une erreur de conception que d'une volonté de ralentir délibérément le transfert des pièces vers la machine située en aval. Néanmoins, il existe quelques situations particulières pour lesquelles ce cas de figure résulte d'un impératif lié au processus de production. Nous pouvons en effet citer des contraintes de temps de séchage ou de refroidissement imposées lors du transfert des pièces. Il est aussi envisageable de prendre en considération une exigence technique particulière telle que la nécessité de respecter un déplacement lent de pièces fragiles ou ne disposant pas d'une fixation suffisante pour permettre un transfert rapide.

De manière intuitive et en considérant le couplage des machines M_{i-1} et M_i , nous pouvons aisément admettre que le comportement en rafales provient du fait que l'espace inter-machines se remplit plus vite qu'il ne se vide. Aussi, lorsqu'il atteint son seuil de saturation, l'activité de la machine M_{i-1} située en amont est suspendue au terme du traitement de la pièce détenue jusqu'à ce que la première pièce en attente soit prise en charge par la machine M_i située en aval.

Il ne semble pas qu'il existe dans la littérature de nombreux résultats théoriques permettant d'évaluer la cadence de production résultant d'une telle situation appliquée à un système flow-shop. Toutefois, quelques publications sont orientées vers ce type de travaux [Yamazaki et al., 1978] [Yamazaki et al., 1985] [Yamazaki et al., 1992]. Le plus souvent, elles tiennent compte de la limitation de la capacité d'encours entre deux machines successives et plus rarement de la vitesse de transfert. De manière générale, la problématique est abordée au moyen de la théorie des files d'attente [Kleinrock, 1975] [Baynat, 2000].

Nous pouvons toutefois déterminer la cadence de production imposée par une phase de transfert en appliquant la classique loi de Little [Little, 1961] couramment utilisée dans le cadre d'analyse de performance des systèmes de file d'attente :

$$L = \lambda W \quad (3.5)$$

Le nombre de clients et le temps passé dans la file d'attente sont respectivement donnés par L et W . Le paramètre λ représente le rythme moyen d'arrivée des clients dans la file d'attente. Le débit du système se note μ et correspond au rythme moyen de sortie des clients par unité de temps. Le fonctionnement propre d'une file d'attente et la règle de conservation des débits moyens qui en découle impliquent que les valeurs des paramètres λ et μ soient égales.

Le délai moyen d'inter-sorties fixée par une phase de transfert entre deux machines successives peut alors être simplement déterminée à l'aide de la loi de Little. En effet et par analogie, la capacité d'encours $\varpi_{i-1,i}$ et le temps $\psi_{i-1,i}$ nécessaire à l'acheminement d'une pièce entre les machines M_{i-1} et M_i sont respectivement le nombre de clients et le temps passé dans le système de file d'attente considéré. La figure 3.2 illustre la phase de transfert entre deux machines successives M_{i-1} et M_i sous la forme d'un système de file d'attente.

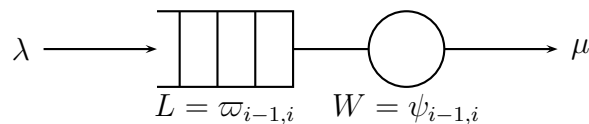


FIG. 3.2: *Système de file d'attente d'une phase de transfert*

Lorsque la cadence de production est fixée par une phase de transfert, elle correspond

à l'inverse du débit du système de file d'attente équivalent. Dans un tel cas, nous retenons alors le cas le plus défavorable du système de production :

$$\tau = \max_{M_i \in [M_2, M_k]} \left\{ \frac{\psi_{i-1,i}}{\varpi_{i-1,i}} \right\} \quad (3.6)$$

En complément, nous pouvons relever que la taille des rafales est conditionnée par le nombre de pièces en attente et donc par la capacité d'encours $\varpi_{i-1,i}$ de l'espace inter-machines constituant le goulot d'étranglement. La machine M_i est en effet normalement alimentée par une succession de $\varpi_{i-1,i}$ pièces à la vitesse de $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ avant d'observer un laps de temps d'attente lié à la durée pendant laquelle l'espace inter-machines a été saturé. Nous pouvons également indiquer que la période des rafales correspond à la durée $\psi_{i-1,i}$ de transfert des pièces entre les machines M_{i-1} et M_i . Il est ainsi possible de généraliser l'évaluation de la cadence de production d'un système flow-shop classique en admettant que le goulot d'étranglement est obtenu en comparant les ratios du temps opératoire sur la capacité de chacune des ressources et en considérant les phases de transfert comme des ressources. Identiquement, la taille des rafales est donnée par la capacité de la ressource constituant le goulot d'étranglement.

La figure 3.3 présente un modèle d'abstraction de la phase de transfert de pièces incluant le convoyeur et les machines adjacentes M_{i-1} et M_i . Chaque portion inter-machines du système flow-shop peut ainsi être représentée sous la forme d'une machine particulière, marquée par une flèche sur la figure, dont le temps opératoire est donné par le ratio du temps de transfert sur la capacité d'encours.

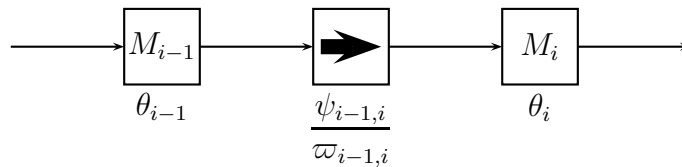


FIG. 3.3: *Modèle d'abstraction d'une phase de transfert*

Nous pouvons alors associer les équations 3.4 et 3.6 permettant de déterminer la cadence de production d'un système flow-shop classique qu'elle soit respectivement fixée par la machine la plus lente ou par la phase de transfert des pièces la plus défavorable. Elle correspond alors à la pire des deux bornes. Dans le second cas, le délai d'inter-fins

de travaux n'est généralement pas constant :

$$\tau = \max \left\{ \max_{M_i \in \mathcal{M}} \{\theta_i\}, \max_{M_i \in [M_2, M_k]} \left\{ \frac{\psi_{i-1,i}}{\varpi_{i-1,i}} \right\} \right\} \quad (3.7)$$

Il est important de noter que, lorsque la capacité d'encours $\varpi_{i-1,i}$ est limitée à une seule pièce, le modèle d'abstraction se réduit à un cas simple où la phase de transfert peut être considérée comme une machine standard. Dans ce contexte particulier, le délai d'inter-fins de travaux respecte une période stricte et aucun comportement en rafales n'est constaté.

3.2.5 Calcul du makespan

Le makespan C_{max} d'un système de production est directement lié à sa cadence de production τ , soit le temps de rétention $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ sur la machine la plus lente dans le cas courant d'un système flow-shop classique, et au nombre n de pièces à produire. Son calcul intègre évidemment le délai de réalisation du premier travail correspondant à la durée minimale de séjour C_{min} d'une pièce sur le système :

$$C_{max} = C_{min} + ((n - 1) \times \max_{M_i \in \mathcal{M}} \{\theta_i\}) \quad (3.8)$$

Quand la cadence de production τ est conditionnée par une phase de transfert, elle représente une valeur supérieure à celle du délai de rétention $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ sur la machine la plus lente. Le délai moyen d'inter-fins de travaux, dégradé par rapport à la borne sur la cadence de production fixée par la machine la plus lente, est vérifié durant de chaque période $\psi_{i-1,i}$. Le calcul de la date d'achèvement d'un travail J_n à l'aide de l'équation 3.8 ne permet alors plus d'obtenir un résultat cohérent pour tous les travaux. En effet, au cours d'une rafale, la production se déroule à la vitesse d'un travail toutes les $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ unités de temps puis un délai est observé entre la date d'achèvement du dernier travail d'une rafale et celle du premier travail de la rafale suivante. Ce délai est lié au nombre $\varpi_{i-1,i}$ de travaux traités au cours d'une rafale et à l'écart de temps entre le délai moyen d'inter-fins de travaux et la durée opératoire de la machine la plus lente $\max_{M_i \in \mathcal{M}} \{\theta_i\}$. Il peut être perçu comme une compensation de l'accélération du système durant les rafales en produisant à la vitesse de la machine la plus lente.

Il est donc nécessaire d'ajouter à l'équation 3.8 une valeur, que nous notons σ , dont le calcul est basé sur le retard de fin d'achèvement du dernier travail J_n de la production par rapport à un système flow-shop classique dont la cadence de production est fixée par la machine la plus lente. Il s'agit alors d'une généralisation du calcul du makespan intégrant le nombre d'occurrences du délai marqué entre deux rafales successives relevé jusqu'à la date d'achèvement du dernier travail J_n :

$$C_{max} = C_{min} + ((n - 1) \times \max_{M_i \in \mathcal{M}} \{\theta_i\}) + \sigma \quad (3.9)$$

Nous pouvons considérer la valeur σ comme un retard établi par la prise en compte de deux paramètres que nous désignons par σ_1 et σ_2 . Le paramètre σ_1 correspond au temps cumulé de compensation calculé en fonction du nombre $\varpi_{i-1,i}$ de travaux traités au cours d'une rafale et de l'écart entre la cadence de production τ et le délai de rétention $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ des pièces sur la machine la plus lente du système. Le paramètre σ_2 représente le nombre de phases inter-rafales constatées au cours l'exécution du système et indique alors le nombre de fois où l'accélération des rafales a été compensée. Il est basé sur le ratio du nombre n de travaux à réaliser sur le nombre $\varpi_{i-1,i}$ de travaux traités au cours d'une rafale :

$$\text{avec } \varpi_{i-1,i} \text{ issu de } \max_{M_i \in [M_2, M_k]} \left\{ \frac{\psi_{i-1,i}}{\varpi_{i-1,i}} \right\} : \sigma \begin{cases} \sigma_1 = (\tau - \max_{M_i \in \mathcal{M}} \{\theta_i\}) \times \varpi_{i-1,i} \\ \sigma_2 = \left\lceil \frac{n}{\varpi_{i-1,i}} \right\rceil - 1 \end{cases} \quad (3.10)$$

Le retard σ résultant s'obtient en multipliant entre-eux les deux paramètres σ_1 , et σ_2 . La valeur du premier paramètre annule naturellement le retard lorsque la cadence de production τ du système considéré est fixée par le temps de rétention $\max_{M_i \in \mathcal{M}} \{\theta_i\}$ sur la machine la plus lente :

$$\sigma = \sigma_1 \times \sigma_2 \quad (3.11)$$

Dans le cadre de nos travaux et compte tenu de la particularité des systèmes flow-shop pour lesquels le goulot d'étranglement se situe au niveau d'une phase de transfert des pièces, nous ne considérons pas spécifiquement ce cas de figure.

3.2.6 Latence et périodicité d'activité des machines

Le délai moyen d'inter-fins de travaux peut être assimilé à une période globale du système s'appliquant communément à l'ensemble des machines. De façon générale, toutes les machines sont en effet soumises à cette même période d'activité. Au cours du régime stationnaire, le délai séparant les dates de démarrage β_{j-1}^i et β_j^i de la réalisation de deux travaux consécutifs J_{j-1} et J_j sur une machine M_i est constant et toujours égal au délai moyen d'inter-fins de travaux, équivalent à la cadence de production τ , lorsque le goulot d'étranglement est constitué par la machine la plus lente du système. :

$$\forall J_j \in [J_2, J_n], \forall M_i \in \mathcal{M} : \tau = \beta_j^i - \beta_{j-1}^i \quad (3.12)$$

La cadence de production est alors imposée par la machine la plus lente dont l'activité est permanente puisqu'elle enchaîne les travaux sans marquer de temps d'attente. Dans ce contexte, le délai d'inter-fins de travaux est constant si un comportement stable et déterministe du système est admis. La date β_1^i de début du premier travail sur une machine M_i , marquant son entrée en production, s'obtient aisément par le cumul des durées de rétention des pièces sur l'ensemble des machines situées en aval et des temps de transfert :

$$\forall M_i \in [M_2, M_k] : \beta_1^i = \sum_{j=1}^{i-1} \theta_j + \sum_{j=2}^{i-1} \psi_{j-1,j} \quad (3.13)$$

La date d'entrée en production correspond naturellement à l'instant 0 pour la première machine et nous avons donc toujours $\beta_1^1 = 0$.

Lorsque la ressource menante est une machine, la date de début de réalisation d'un travail J_j sur une machine M_i se détermine à partir de la période globale du système, correspondant à la cadence de production τ , et de la date β_1^i de début de réalisation du premier travail sur la machine concernée. La date β_j^i s'obtient à l'aide de l'équation 3.13 :

$$\forall M_i \in \mathcal{M}, \forall J_j \in \mathcal{J} : \beta_j^i = \beta_1^i + ((j-1) \times \max_{M_i \in \mathcal{M}} \{\theta_i\}) \quad (3.14)$$

Identiquement à l'évaluation du makespan, il est nécessaire d'ajuster le calcul de la date β_j^i de début de réalisation d'un travail J_j sur une machine M_i afin de considérer le cas général en intégrant le retard σ donné par les équations 3.10 et 3.11. Il est en effet parfois nécessaire de prendre en compte le contexte de systèmes pour lesquels le délai

moyen d'inter-fins de travaux est fixé par une phase de transfert de pièces :

$$\forall M_i \in \mathcal{M}, \forall J_j \in \mathcal{J} : \beta_j^i = \beta_1^i + ((j-1) \times \max_{M_i \in \mathcal{M}} \{\theta_i\}) + \sigma \quad (3.15)$$

Compte tenu du caractère périodique des systèmes, il est alors possible de connaître a priori la date β_j^i de début de réalisation de tout travail J_j sur chacune des machines du système flow-shop étudié. Les dates ξ_j^i de fin de réalisation des travaux découlent naturellement des équations 3.14 et 3.15 en ajoutant le temps d'exécution θ_i du cycle opératoire de la machine M_i concernée.

La latence d'activité d'une machine est le délai qui s'écoule entre la date de fin d'un travail, mais pas nécessairement le relâchement de la pièce associée, et la date de début du travail suivant. Nous la notons $\bar{\theta}_i$ par opposition à θ_i relatant l'activité de la machine M_i pour la réalisation d'un travail. La latence se détermine de manière évidente par la différence entre la période globale du système, soit la cadence de production τ , et la durée de rétention minimale des pièces sur la machine M_i :

$$\forall M_i \in \mathcal{M} : \bar{\theta}_i = \tau - \theta_i \quad (3.16)$$

Selon la position d'une machine M_i donnée sur le système par rapport à la plus lente, sa latence d'activité se traduit de deux façons. Dans le premier cas, elle consiste en une suspension d'activité conduisant à une attente entre la réalisation de deux travaux consécutifs. Dans le second cas, la latence d'activité se présente sous la forme d'un allongement du délai de présence des pièces sur la machine due à l'impossibilité de les libérer dès la fin de leur traitement en raison d'une saturation de l'espace d'encours entre la machine M_i et la suivante.

3.2.7 Exemples de systèmes simulés

Nous présentons dans cette section deux exemples d'exécution de systèmes flow-shop classiques intégrant cinq machines. Ils sont différenciés par l'équipement critique constituant le goulot d'étranglement. Le premier exemple décrit un cas de figure couramment rencontré pour lequel le délai d'inter-fins de travaux est fixé par la machine la plus lente du système. Le second exemple s'applique à un système flow-shop dont la cadence est définie par une phase de transfert des pièces entre deux machines. La gamme opératoire

se décompose en un jeu de cinq opérations et chaque machine n'exécute alors qu'une seule opération.

Pour chacun des deux exemples présentés, l'exécution du système flow-shop étudié est graphiquement représentée sous forme d'un diagramme de Gantt sur les cinquante premières unités de temps d'activité. La présence d'une pièce sur une machine est marquée par un rectangle gris. Les phases de suspension d'activité des machines observées durant la rétention d'une pièce sont indiquées par un trait horizontal. L'ensemble des dates de début β_j^i et de fin ξ_j^i de travaux sur chacune des machines est aussi récapitulé dans un tableau.

3.2.7.1 Cadence fixée par la machine la plus lente

Pour ce premier cas et selon l'équation 3.7, la cadence moyenne de production τ est fixée par la machine la plus lente du système. Il s'agit de la seconde machine dont la durée opératoire θ_2 est de quatre unités de temps. Nous avons donc dans cet exemple $\tau = \theta_2$. Le tableau 3.1 présente l'ensemble des données statiques définissant le paramétrage du système flow-shop considéré.

M_i	θ_i	$\bar{\theta}_i$	$\psi_{i-1,i}$	$\varpi_{i-1,i}$	$\frac{\psi_{i-1,i}}{\varpi_{i-1,i}}$	τ
1	2	2	-	-	-	4
2	4	0	1	3	0.33	
3	1	3	2	4	0.5	
4	2	2	1	2	0.5	
5	3	1	3	3	1	

TAB. 3.1: Données de paramétrage du premier exemple

La figure 3.4 montre le diagramme de Gantt d'enchaînement des opérations. De manière visuelle, nous pouvons estimer que le régime stationnaire du système coïncide avec l'entrée en production de la dernière machine, soit à la seizième unité de temps. La figure permet de constater que les délais d'inter-débuts et d'inter-fins de travaux sur chacune des machines sont constants en régime stationnaire et respectent toujours la durée opératoire θ_2 de la seconde machine qui constitue le goulot d'étranglement. Seule la première machine, située en amont de la ressource menante, est soumise à des phases de suspension d'activité. La capacité d'encours de travaux $\varpi_{1,2}$ fixée à trois pièces contraint

la première machine à adapter le rythme de son activité à la période globale du système, soit la cadence de production τ , à partir de la fin d'exécution des opérations pour le septième travail à l'instant 14. Les quatrième et sixième travaux saturent alors l'encours qui atteint la valeur limite $\varpi_{1,2}$. La première machine de production du système retient alors la pièce associée dont le relâchement n'est déclenché qu'à l'instant 15, ce qui lui permet de reprendre son activité et de démarrer une nouvelle occurrence de son cycle opération pour le traitement du huitième travail.

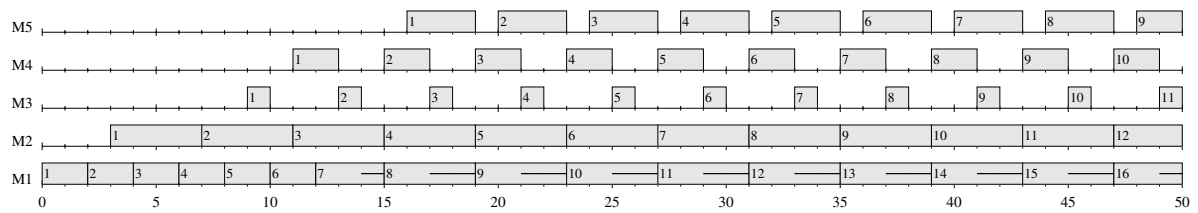


FIG. 3.4: Diagramme d'exécution du premier exemple

Le tableau 3.2 présente des données dynamiques du système flow-shop étudié. Il montre que l'exécution se déroule de manière strictement périodique. L'activité de chacune des machines respecte le rythme de réalisation d'un travail au cours d'un cycle opératoire de période τ .

$\frac{\beta_j^i}{\xi_j^i}$		J_j									
		1	2	3	4	5	6	7	8	9	10
M_i	1	$\frac{0}{2}$	$\frac{2}{4}$	$\frac{4}{6}$	$\frac{6}{8}$	$\frac{8}{10}$	$\frac{11}{13}$	$\frac{15}{17}$	$\frac{19}{21}$	$\frac{23}{25}$	$\frac{27}{29}$
	2	$\frac{3}{7}$	$\frac{7}{11}$	$\frac{11}{15}$	$\frac{15}{19}$	$\frac{19}{23}$	$\frac{23}{27}$	$\frac{27}{31}$	$\frac{31}{35}$	$\frac{35}{39}$	$\frac{39}{43}$
	3	$\frac{9}{10}$	$\frac{13}{14}$	$\frac{17}{18}$	$\frac{21}{22}$	$\frac{25}{26}$	$\frac{29}{30}$	$\frac{33}{34}$	$\frac{37}{38}$	$\frac{41}{42}$	$\frac{45}{46}$
	4	$\frac{11}{13}$	$\frac{15}{17}$	$\frac{19}{21}$	$\frac{23}{25}$	$\frac{27}{29}$	$\frac{31}{33}$	$\frac{35}{37}$	$\frac{39}{41}$	$\frac{43}{45}$	$\frac{47}{49}$
	5	$\frac{16}{19}$	$\frac{20}{23}$	$\frac{24}{27}$	$\frac{28}{31}$	$\frac{32}{35}$	$\frac{36}{39}$	$\frac{40}{43}$	$\frac{44}{47}$	$\frac{48}{51}$	$\frac{52}{55}$

TAB. 3.2: Dates de début et de fin de travaux sur les machines du premier exemple

Les enchaînements de travaux sur le système représenté sont exclusivement condi-

tionnés par la vitesse de traitement des pièces par la seconde machine de production. Le délai de réalisation du premier travail, correspondant au temps de présence minimal C_{min} des pièces sur le système donné par l'équation 3.1, est de 19 unités de temps et peut être vérifié graphiquement. Il représente la durée effective de traitement et de transfert des pièces sur le système. Le makespan C_{max} est établi à 55 unités de temps pour un ensemble de dix travaux. Son évaluation peut être conjointement confirmée par les équations 3.8 et 3.9.

3.2.7.2 Cadence fixée par une phase de transfert

Ce second cas reprend les données statiques du premier exemple auxquelles un seul changement a été apporté. Le temps de transfert des pièces $\psi_{3,4}$ entre les troisième et quatrième machines est élevé à dix unités de temps. Conformément à la valeur calculée par l'équation 3.7, le délai d'inter-fins de travaux, qui dans ce cas est une valeur moyenne, est porté à cinq unités de temps. Le goulot d'étranglement se situe alors au niveau de la phase de transfert entre les troisième et quatrième machines du système. Dans ce cas, la ressource critique du système ne correspond pas à une machine de production. Nous avons alors pour cet exemple une cadence moyenne $\tau = \frac{\psi_{3,4}}{\varpi_{3,4}}$. Le tableau 3.3 présente les principales données statiques du système représenté.

M_i	θ_i	$\bar{\theta}_i$	$\psi_{i-1,i}$	$\varpi_{i-1,i}$	$\frac{\psi_{i-1,i}}{\varpi_{i-1,i}}$	τ
1	2	2	-	-	-	5
2	4	0	1	3	0.33	
3	1	3	2	4	0.5	
4	2	2	10	2	5	
5	3	1	3	3	1	

TAB. 3.3: Données de paramétrage du second exemple

Le diagramme de Gantt de la figure 3.5 permet de constater que la production s'effectue de manière périodique et respecte une succession de rafales de deux pièces, soit la capacité d'encours $\varpi_{3,4}$ entre les troisième et quatrième machines du système. Le régime stationnaire semble s'établir dès l'entrée en production de la dernière machine. Au cours de chaque rafale, la cadence de production du système est régulière et un travail s'achève toutes les quatre unités de temps correspondant à la durée opératoire θ_2

de la seconde machine. En régime stationnaire, l'espace d'encours $\varpi_{3,4}$ rencontre des phases de saturation régulières et alternées. Ce comportement provoque la suspension d'activité de la troisième machine durant successivement trois et cinq unités de temps retardant ainsi le relâchement des pièces au terme de la fin d'exécution de son cycle opératoire.

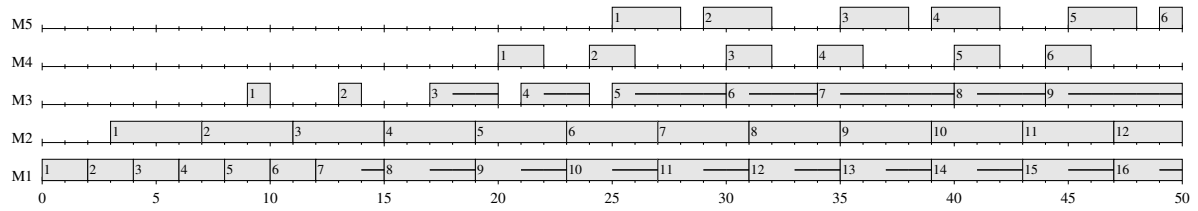


FIG. 3.5: Diagramme d'exécution du second exemple

Le tableau 3.4 présente les données résultant de l'exécution du système étudié. Le temps de présence minimal C_{min} des pièces sur le système est de 28 unités de temps. Il correspond au délai de réalisation du premier travail. Le makespan C_{max} est porté à 72 unités de temps pour la production d'un lot de dix pièces. Son calcul est confirmé par l'équation 3.9.

$\frac{\beta_j^i}{\xi_j^k}$		J_j									
		1	2	3	4	5	6	7	8	9	10
M_i	1	$\frac{0}{2}$	$\frac{2}{4}$	$\frac{4}{6}$	$\frac{6}{8}$	$\frac{8}{10}$	$\frac{10}{12}$	$\frac{12}{14}$	$\frac{15}{17}$	$\frac{19}{21}$	$\frac{23}{25}$
	2	$\frac{3}{7}$	$\frac{7}{11}$	$\frac{11}{15}$	$\frac{15}{19}$	$\frac{19}{23}$	$\frac{23}{27}$	$\frac{27}{31}$	$\frac{31}{35}$	$\frac{35}{39}$	$\frac{39}{43}$
	3	$\frac{9}{10}$	$\frac{13}{14}$	$\frac{17}{18}$	$\frac{21}{22}$	$\frac{25}{26}$	$\frac{30}{31}$	$\frac{34}{35}$	$\frac{40}{41}$	$\frac{44}{45}$	$\frac{50}{51}$
	4	$\frac{20}{22}$	$\frac{24}{26}$	$\frac{30}{32}$	$\frac{34}{36}$	$\frac{40}{42}$	$\frac{44}{46}$	$\frac{50}{52}$	$\frac{54}{56}$	$\frac{60}{62}$	$\frac{64}{66}$
	5	$\frac{25}{28}$	$\frac{29}{32}$	$\frac{35}{38}$	$\frac{39}{42}$	$\frac{45}{48}$	$\frac{49}{52}$	$\frac{55}{58}$	$\frac{59}{72}$	$\frac{65}{68}$	$\frac{69}{72}$

TAB. 3.4: Dates de début et de fin de travaux sur les machines du second exemple

Enfin, nous pouvons également relever que le délai séparant les dates début de deux rafales consécutives respecte bien une période de dix unités de temps, soit $\psi_{3,4}$. Ce cycle

se constate aussi au niveau de l'activité de la troisième et de la quatrième machine du système modélisé.

3.3 Mise en œuvre d'un contrôleur central

Dans la section précédente, nous avons présenté les caractéristiques de base des lignes d'assemblage automatisées que nous étudions. Cependant, un aspect particulier des systèmes flow-shop que nous considérons réside dans l'utilisation d'un contrôleur pour conditionner et coordonner le cycle des machines. L'activité du contrôleur est basée sur le traitement de messages d'information nécessaires au pilotage des machines.

Dans cette section, nous apportons alors une extension au modèle de base préalablement défini afin d'y intégrer les éléments relatifs à la mise œuvre d'un contrôleur central.

3.3.1 Notions de contrôle et de commande

D'une façon générale, un système automatisé conçu pour répondre à un besoin de production manufacturière implique la mise en œuvre de calculateurs et d'un traitement d'informations diverses. La gestion d'un tel système est répartie sur deux niveaux [Aléonard et al., 1994] [Trentesaux, 1996]. Le niveau supérieur est celui de la *gestion prévisionnelle* qui consiste à anticiper l'activité de production et à programmer un ensemble d'actions ou de décisions en fonction de la demande exercée sur la quantité et le type des pièces à produire. Le niveau inférieur s'applique au *pilotage* des équipements. Il élabore le plan d'exécution à court terme des opérations exécutées par les machines en réagissant aux événements liés à la dynamique et l'état courant du système de production.

Le cadre de nos travaux s'applique au niveau du pilotage de la gestion des systèmes de production. La fonction de pilotage comprend deux sous-fonctions que sont la *conduite* et la *commande* [Archimède, 1991] [Dindeleux, 1992] [Tchako, 1994]. La commande est chargée des activités de pilotage de bas niveau. Elle est destinée à interagir avec les capteurs et actionneurs du système de production. La conduite, correspondant au niveau décisionnel du pilotage, a pour rôle de mettre en œuvre les décisions planifiées au niveau de la gestion prévisionnelle. La conduite constitue ainsi une sous-fonction du système de

pilotage plus élevée que la commande à laquelle elle impose des décisions. Elle intègre principalement des activités de *contrôle* assurant la cohérence globale du système de pilotage et la coordination des différentes unités de commande.

Dans le domaine de l'automatisation des procédés industriels, le terme *contrôle-commande* est le plus souvent utilisé pour décrire les activités de pilotage de bas niveau.

3.3.2 Structure du système de pilotage

Les modes de pilotage de procédés industriels sont généralement classés en fonction de l'architecture matérielle et logicielle du système d'information qui les supporte. Ils peuvent être centralisés, hiérarchisés ou distribués [Dilts et al., 1991] [Veeramani, 1994]. Dans le cas d'une structure centralisée, les capteurs et actionneurs de la partie opérative sont directement lus et commandés par un équipement unique. Une structure est hiérarchisée lorsque le système de pilotage est décomposé en plusieurs niveaux. Un système de pilotage est distribué quand il concerne un ensemble d'entités indépendantes mais s'échangeant des informations dans le but de coordonner leurs actions. Des architectures hybrides ont également été introduites [Trentesaux, 1996].

La figure 3.6 montre un exemple de système de pilotage basé sur le modèle que nous considérons. Il inclut un contrôleur et un ensemble d'unités de commande placées sur trois machines notées M_1 , M_2 et M_3 . Toutefois, nous représentons également les capteurs et actionneurs en raison de leur intégration au circuit d'informations. Les arcs orientés représentent le sens de circulation des informations.

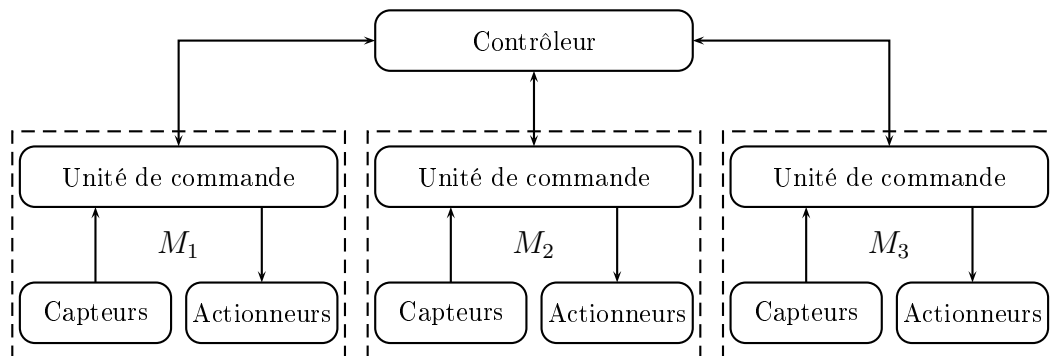


FIG. 3.6: Organisation hiérarchisée du système de pilotage

Du point de vue de sa structure, le système de pilotage s'appuie sur une organisation

hiérarchisée sur deux niveaux. Le niveau bas regroupe ainsi l'ensemble des unités de commande en relation avec les capteurs et actionneurs et le niveau haut est occupé par le contrôleur. Dans ce type de structure, un équipement de niveau supérieur coordonne les activités d'un groupe d'équipements de niveau inférieur selon une relation maître-esclave [Jones et al., 1989]. Il n'existe aucune possibilité de dialogue direct entre les différentes unités de commande. Le contrôleur centralise et distribue l'intégralité des informations nécessaires au contrôle du système de production.

3.3.3 Charge du contrôleur et interactions avec les machines

Par le biais de leur unité de commande, les machines émettent des requêtes sous forme de messages au contrôleur à plusieurs reprises au cours de leur cycle opératoire. Après l'envoi d'un message de requête vers le contrôleur, les machines suspendent leur activité. Le contrôleur est alors chargé de traiter la requête et de produire une réponse qui est envoyée vers la machine émettrice qui peut ainsi reprendre son cycle opératoire. Contrairement au mode d'interaction maître-esclave typique dans un système de pilotage hiérarchisé où les échanges sont déclenchés par l'équipement de niveau supérieur, le système de pilotage que nous considérons est basé sur des échanges toujours initiés par les équipements de niveau inférieur que sont les unités de commande des machines de production. Il s'agit alors d'un mode d'interaction client-serveur impliquant des échanges sous forme de requêtes et de réponses entre différents processus communicants [Gardarin and Gardarin, 1996].

Les messages de requête sont porteurs de différents types d'informations. Dans un contexte d'application industrielle, ils concernent dans la plupart des cas une demande du statut de la pièce détenue ou une demande de consigne afin de paramétrer une opération mécanique. Les messages de requête sont aussi couramment utilisés pour remonter vers le contrôleur des données diverses destinées à être centraliser puis redistribuer à des fins de coordination des machines de production. Ils sont également utilisés pour répondre à une nécessité d'archivage d'informations liées à la traçabilité de la production.

Les messages de requête peuvent être assimilés à des opérations particulières entrant dans la gamme de fabrication d'une pièce. Ils se distinguent des opérations mécaniques, non seulement par leur nature informationnelle, mais également par la localisation de leur exécution. Les messages sont donc des opérations affectées à une machine de pro-

duction mais dont l'exécution est ensuite confiée au contrôleur. Le délai minimal θ_i de rétention d'une pièce sur une machine M_i n'est donc pas uniquement établi par la somme des délais d'exécution d'une série d'opérations mécaniques exécutées localement, il intègre également un temps cumulé de traitements distants de messages de requête confiés au contrôleur central. Afin de distinguer les opérations locales et distantes, nous étendons la notation en ajoutant un astérisque en exposant pour marquer les traitements de requête inclus dans le cycle opératoire des machines de production. De même et plus globalement, la notation θ_i^* exprime la part de charge du contrôleur pour la réalisation d'un travail sur la machine de production M_i .

La figure 3.7 détaille un exemple du déroulement de la réalisation d'un travail sur une machine M_i . La durée opératoire θ_i nécessaire à la réalisation d'un travail correspond à un ensemble de cinq opérations locales et distantes respectivement marquées en gris et blanc. Les deuxième et quatrième opérations sont des traitements de requête exécutés par le contrôleur. Par conséquent, nous avons $\mathcal{O}_i = \{O_{i,1}, O_{i,2}^*, O_{i,3}, O_{i,4}^*, O_{i,5}\}$ pour l'exemple présenté avec $\theta_i^* = \theta_{i,2}^* + \theta_{i,4}^*$. La phase d'attente d'un message dans le tampon de communication du contrôleur avant son traitement n'est pas systématique et sa durée dépend évidemment du plan de charge du contrôleur avant que le message ne soit sélectionné. Les phases d'attente sont représentées sur la figure par des zones hachurées. Le temps de réponse du contrôleur à une requête émise par la machine correspond au temps symbolisé par la distance entre une flèche montante et la flèche descendante suivante.

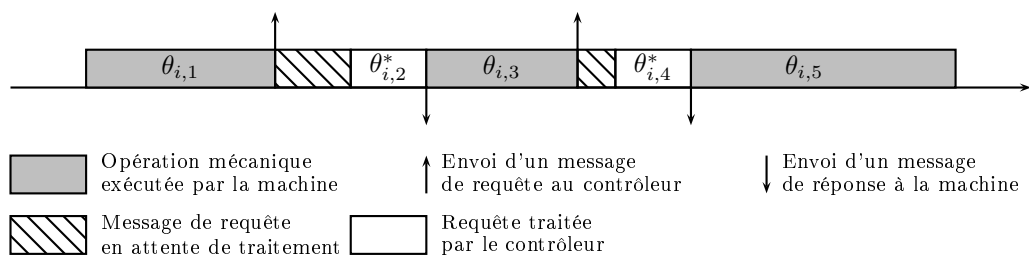


FIG. 3.7: Exemple de chronogramme de réalisation d'un travail sur une machine

Tout comme les machines de production, le contrôleur central est une ressource disjonctive qui se limite au traitement d'un seul message de requête à la fois. La concurrence potentielle d'accès au contrôleur due à l'activité décorellée des machines et son incapacité à traiter plusieurs messages simultanément nécessitent la mise en œuvre d'un

tampon de communication pour accueillir et stocker les messages préalablement à leur traitement.

La figure 3.8 présente le modèle des systèmes flow-shop contrôlés que nous considérons. Le contrôleur est désigné par la lettre C . Il constitue une ressource partagée par l'ensemble des machines de production dont l'accès s'effectue ainsi en exclusion mutuelle. Pour la réalisation complète d'un travail de la première à la dernière machine, la charge du contrôleur s'exprime par la somme des temps d'exécution des messages de requête émis par l'ensemble des machines de production, soit $\sum_{M_i \in \mathcal{M}} \theta_i^*$.

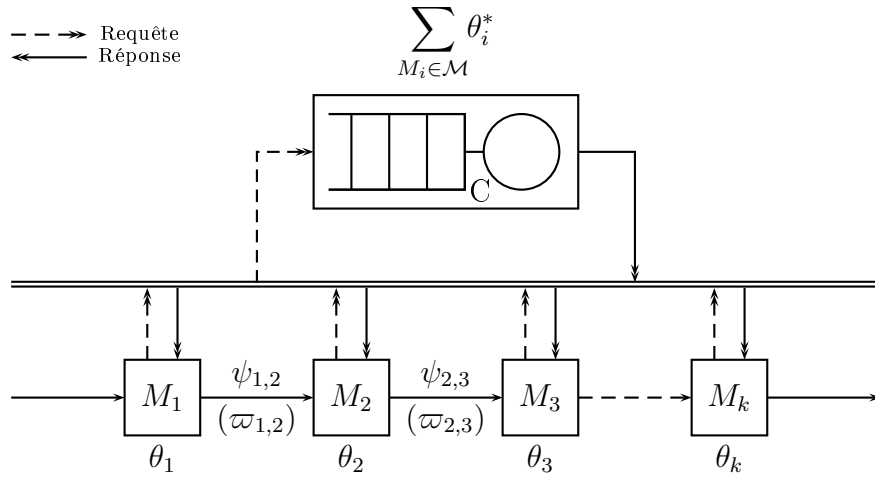


FIG. 3.8: *Modèle des systèmes contrôlés*

Les temps de transmission des messages de requête depuis les unités de commande des machines de production vers le contrôleur central peuvent être assimilés à des délais de transfert des pièces entre deux machines successives. Dans un contexte de réalité industrielle, ces échanges sont généralement supportés par un réseau de terrain [Rubino and Toutain, 1998] [Thomesse, 2004] [Thomesse, 2006]. Nous supposons cependant que le délai de propagation d'un message de requête entre une unité de commande et le contrôleur est de durée nulle. Cette hypothèse simplificatrice est principalement motivée par le fait que la durée de transmission d'un message de requête est considérablement plus courte que le temps nécessaire à son traitement par le contrôleur. Elle est également négligeable en comparaison avec la durée d'exécution des opérations mécaniques assurées par les machines de production ou de transfert des pièces entre deux machines successives.

Un aspect important de l'activité du contrôleur réside dans le mécanisme de préemption qu'il peut implémenter. En effet et contrairement aux opérations mécaniques exécutées par les machines de production, le traitement des messages peut être aisément interrompu puis repris ultérieurement afin de répondre à une gestion par priorités des requêtes émises par les unités de commande.

3.4 Expression de la problématique

La mise en œuvre d'un contrôleur central pour assurer le pilotage des systèmes de fabrication considérés par notre étude agit sur le comportement global et la performance des équipements contrôlés. Son activité consiste à servir les différentes machines de la partie opérative du système flow-shop en excluant les possibilités de traitements simultanés de messages de requête.

En raison du comportement déterminisme des systèmes flow-shop considérés dans nos travaux, l'évaluation d'une solution d'ordonnancement des messages peut être envisagée selon une méthode hors ligne. Toutefois, dans un contexte industriel pratique, il est préférable d'effectuer la sélection des messages durant l'exécution du système afin de mieux répondre aux perturbations pouvant agir sur son comportement. En effet, le choix d'un mécanisme de sélection en ligne permet de réagir en temps réel aux éventuelles dérives du délai de rétention des pièces sur les machines et de compenser certains aléas de fonctionnement.

3.4.1 Contrainte sur le temps de réponse du contrôleur

Avec la mise en œuvre d'un contrôleur central, la durée de rétention d'une pièce sur une machine M_i ne se limite plus à la durée du cycle opératoire θ_i . L'évaluation du temps de réponse à une requête émise par une unité de commande doit en effet prendre en compte un délai d'attente variable que les messages doivent observer dans le tampon de communication avant le démarrage de leur traitement. Ainsi pour une machine M_i , le cumul des temps de réponse aux différentes requêtes nécessaires à la réalisation d'un travail donné est de $\theta_i^* + \delta$ où δ représente la durée cumulée de séjour des messages dans le tampon de communication. La durée de rétention d'une pièce sur une machine M_i devient alors $\theta_i + \delta$.

Afin de garantir la productivité des systèmes de production étudiés, il est nécessaire de maîtriser le temps de réponse aux requêtes et de s'efforcer de le minimiser. Le travail exposé dans cette thèse s'attache donc à établir une règle de sélection des messages et à ajuster la coordination de l'activité des machines de production par le contrôleur afin d'obtenir le meilleur délai d'inter-fins de travaux et de satisfaire au mieux les objectifs de production.

3.4.2 Spécification du problème d'ordonnancement

La spécification du système de production contrôlé ne peut être directement effectuée par la notation de Graham. En effet, nous ne pouvons exprimer deux caractéristiques essentielles du modèle que nous considérons.

Tout d'abord, l'utilisation d'un contrôleur central nécessite d'être renseignée. Les systèmes que nous étudions repose sur le modèle d'un flow-shop classique sur lequel est implantée une ressource partagée par l'ensemble des machines. Le champ α doit préciser cette particularité. De plus, bien que la possibilité de préemption puisse être exprimée par l'indication *pmtn* dans le champ β , elle s'applique à l'ensemble des ressources et des opérations du problème spécifié. Pour le modèle que nous considérons, seules les opérations exécutées par le contrôleur sont interruptibles. Par conséquent, il nous faut préciser que le mécanisme de préemption n'est applicable qu'à la seule activité du contrôleur.

Nous proposons alors deux extensions à la notation de Graham. Pour spécifier la mise en œuvre du contrôleur sous la forme d'une ressource partagée, nous ajoutons l'indication *C1* dans le champ α . Ce choix respecte le formalisme des extensions *S1* et *R1* apportées au champ α pour la prise en compte d'un serveur [Hall et al., 2000] et d'un robot [Hurink and Knust, 2001]. Concernant la possibilité de préemption uniquement accordée au contrôleur, nous précisons par un indice *c* ajouté à l'indication *pmtn* que cette caractéristique ne s'applique que partiellement au système.

Le délai d'inter-fins de travaux traduit la performance du système en termes de productivité et de capacité à atteindre les objectifs de production pour lequel il est conçu. Dans un contexte d'application industrielle, il s'agit donc du critère qu'il est avant tout nécessaire de minimiser s'il est admis que l'accélération des cadences représente l'objectif principal d'amélioration des systèmes de production. Nos travaux vise cet objectif.

En conséquence, l'expression du problème d'ordonnancement correspondant à nos travaux s'écrit $F, C1/pmtn_c, permu, block, r_j/C_{max}$.

3.5 Conclusion

Dans ce chapitre nous exposons le contexte de nos travaux en présentant le modèle des systèmes de production étudiés. La particularité de ces derniers se situe dans la mise en œuvre d'un contrôleur central dont le rôle est de coordonner l'activité des unités de commande. La communication entre les unités de commande et le contrôleur s'effectue au moyen d'échanges de messages porteur d'une requête ou d'une réponse.

Les unités de commande sollicitent le contrôleur à plusieurs reprises au cours de leur cycle opératoire par le biais d'une requête et se placent en attente d'une réponse. Le rythme de leur activité est ainsi conditionné par le temps de réponse aux requêtes. En effet, la concurrence d'accès au contrôleur ne lui permet pas de garantir le traitement des messages dès leur arrivée.

Le travail présenté dans cette thèse consiste alors à optimiser la coordination des machines de production et le mode de consommation des messages par le contrôleur afin de réduire le temps de réponse aux messages de requête et ainsi de minimiser le makespan des systèmes de production modélisés.

Partie II

Étude des systèmes considérés

Chapitre 4

Modélisation des lignes d'assemblage sous la forme d'un réseau de Petri

Sommaire

4.1	Modes d'abstraction des systèmes	82
4.2	Analyse de performances	84
4.2.1	Critères à analyser dans le cadre de l'étude	84
4.2.2	Sélection d'une méthode	85
4.2.3	Détail des phases de modélisation et de simulation	87
4.3	Choix d'un formalisme de modélisation	88
4.3.1	Principaux outils pour l'étude de systèmes à événements discrets	88
4.3.2	Intérêts des réseaux de Petri pour l'analyse de performances des systèmes de production	89
4.4	Description des éléments du modèle	90
4.4.1	Activité d'une machine de production	91
4.4.2	Enchaînement des machines de production	93
4.4.3	Traitement de messages de requête par le contrôleur	94
4.4.3.1	Cas non-préemptif et préemptif	94
4.4.3.2	Sélection de la machine à servir	97
4.4.4	Construction du modèle coloré temporisé	98
4.5	Définition des configurations étudiées	106
4.5.1	Paramétrage initial des machines de production	107
4.5.2	Déclinaisons de configurations pour les systèmes existants	109
4.5.3	Règles de tri évaluées et principal élément de comparaison	110
4.6	Conclusion	111

La première étape de notre travail est d'étudier globalement le comportement des lignes d'assemblage considérées. De manière plus particulière, elle est destinée à analyser l'impact du contrôleur central sur les performances quantitatives des systèmes de production. L'objectif est alors d'évaluer le niveau de dégradation de la cadence de production par rapport aux mêmes systèmes mais pour lesquels toutes les opérations distantes seraient exécutées sans attente par le contrôleur.

Dans la plupart des domaines scientifiques, un système est étudié par l'analyse d'un modèle plutôt que par celle du système lui-même. Selon les caractéristiques des systèmes étudiés, la démarche s'appuie alors sur l'utilisation de méthodes analytiques ou basées sur des phases de simulation. Le degré d'automatisation élevé et la complexité sous-jacente des systèmes de production modernes rendent critiques les étapes de conception et de mise au point. Les erreurs de conception peuvent avoir des conséquences importantes au regard des investissements consentis pour le développement et la réalisation de ces systèmes complexes. Les méthodes analytiques deviennent difficilement applicables à leur évaluation et les modèles doivent alors être étudiés au moyen des techniques conjointes de modélisation et de simulation.

Ce chapitre décrit en premier lieu le contexte d'analyse de performances dans lequel nous nous plaçons. Il justifie ensuite le choix du formalisme des réseaux de Petri avant de détailler la démarche de construction et de paramétrage d'un modèle destiné à l'étude des systèmes que nous considérons. Il s'agit d'étudier le comportement des lignes d'assemblage dans leur globalité en modélisant le cycle opératoire des machines de production, les phases de transfert de pièces et l'activité du contrôleur. Les hypothèses simplificatrices sur lesquelles s'appuie le modèle sont argumentées.

4.1 Modes d'abstraction des systèmes

L'étude du comportement d'un système impose, préalablement à sa modélisation, de définir un mode d'abstraction permettant d'appréhender au mieux sa dynamique et la manière avec laquelle il évolue au cours du temps. Classiquement, trois approches sont à distinguer [Law and Kelton, 1991] [Dang, 2000] [Cassandra et al., 2006] :

- les systèmes continus,
- les systèmes à événements discrets,
- les systèmes hybrides.

Les systèmes continus sont caractérisés par une évolution continue de leur état dans le temps. Dans le cas des systèmes à événements discrets, les changements d'état ne se produisent qu'à certains instants. D'une manière formelle, un système à événements discrets peut être vu comme un système dynamique possédant un espace d'états discrets et un mécanisme de transition d'état [Ho, 1985] [Cassandra et al., 1995]. La dynamique d'un tel système est assurée par l'occurrence des événements se produisant de façon instantanée. En l'absence d'événement, l'état du système demeure inchangé. L'évolution du temps entre deux événements ne provoque aucun effet détectable sur le système. Nous retenons la définition suivante [Ferrier and Boimond, 2004] :

Définition : Un système à événements discrets est un système à espace d'états discrets dont les transitions entre états sont associées à l'occurrence d'événements discrets asynchrones.

Les systèmes physiques issus d'une conception humaine sont fréquemment représentés sous la forme de systèmes à événements discrets. Les changements significatifs d'état sont alors basés sur des occurrences d'événements asynchrones et ponctuels donnant lieu à des phénomènes de parallélisme, de concurrence et de synchronisation. Ce mode d'abstraction est souvent utilisé pour analyser des systèmes obéissant à des règles algorithmiques tels que des systèmes informatiques, des réseaux de communication ou des systèmes de production [Cassandra, 1993]. Un système à événements discrets met en œuvre des actions qui peuvent être considérées comme atomiques. Ce type de systèmes peut donc être décrit par des variables prenant des valeurs symboliques ou logiques. Leur évolution, induit par des événements extérieurs ou des changements spontanés, conditionne la dynamique et le comportement des systèmes modélisés.

L'approche hybride est évoquée pour décrire des systèmes mixant les caractéristiques discrètes et continues. La dynamique repose ainsi conjointement sur des évolutions continues et des transitions discrètes [Barton and Pantelides, 1994] [Zaytoon, 2001].

La modélisation d'un système de production est généralement abordée en fonction de la circulation des produits fabriqués et reproduit alors l'écoulement des flux de matières. Les pièces en cours de production n'évoluent pas de façon continue, mais selon un ensemble fini d'états stables. Les lignes d'assemblage étudiées sont alors perçues comme des systèmes dans lesquels les variables d'état évoluent à des instants discrets. Aussi, l'approche discrète est retenue dans le cadre de ce travail.

4.2 Analyse de performances

Dans le contexte de l'étude des systèmes à événements discrets, deux approches sont généralement considérées [Zhou and Jeng, 1998] [Eldabi et al., 2002] :

- l'approche qualitative,
- l'approche quantitative.

L'approche qualitative s'attache à évaluer la correction logique des systèmes étudiés en vérifiant l'absence de blocages et en validant la succession de certains événements. Elle s'applique ainsi à l'étude de propriétés structurelles et comportementales des systèmes telles que la vivacité et la sûreté. L'approche quantitative, qui constitue le cadre principal de nos travaux, consiste à évaluer le niveau de performances en mesurant un nombre d'événements donnés survenus au cours d'un laps de temps.

Le niveau de performances d'un système dépend de nombreux facteurs qui sont pour la plupart directement liés à la phase de conception. Une analyse de performances est alors menée pour estimer le comportement d'un système existant ou planifié dans le but de procéder à l'évaluation de différentes configurations et de dégager un paramétrage optimal. Il s'agit alors d'une *étude par comparaison*. L'intérêt est aussi de détecter, de comprendre et de résoudre les éventuels problèmes avant la mise en œuvre du système. Il est en effet important de vérifier que le système étudié respecte les conditions et les contraintes qui s'appliquent à son exécution. Dans ce cas, l'analyse correspond à une *étude de dimensionnement* du système.

4.2.1 Critères à analyser dans le cadre de l'étude

Nos travaux s'attachent à évaluer l'impact sur les systèmes de production étudiés de l'exécution d'opérations distantes confiées au contrôleur central par les machines au cours du déroulement de leur cycle opératoire. L'analyse de performances décrite est alors essentiellement abordée dans un contexte de comparaison des principales règles de tri. Cependant, il s'agit aussi d'étudier le dimensionnement des systèmes contrôlés en validant a priori leur niveau de productivité et leur capacité à atteindre une cadence préalablement définie lors de la phase de conception du procédé.

L'objectif de l'analyse de performances est ainsi d'évaluer la pertinence de différentes règles de tri par rapport à la cadence de production et donc au makespan qui constitue

la fonction objectif du problème d'ordonnancement considéré par nos travaux.

En complément, d'autres critères secondaires sont étudiés. Ils concernent essentiellement le taux de charge et le temps de réponse du contrôleur central ainsi que le nombre de messages de requête en attente de traitement dans le tampon de communication.

4.2.2 Sélection d'une méthode

Plusieurs méthodes d'analyse de performances existent. Le choix d'une méthode dépend de différentes caractéristiques du système étudié, telles que sa nature, sa complexité et sa taille, mais également des paramètres à évaluer. De manière courante, trois principales méthodes d'analyse de performances d'un système sont proposées dans la littérature [Jain, 1991] [Wells, 2002] :

- les mesures,
- les méthodes analytiques,
- la simulation.

La méthode par mesures consiste à effectuer des relevés quantitatifs pouvant apporter des réponses précises concernant les performances d'un système existant. Le comportement du procédé est alors directement observé. Cependant, les relevés obtenus ne sont pas toujours le reflet exact des performances réelles. Les mesures peuvent, en effet, être effectuées lors d'une période de charge particulière et occulter certaines situations. Ainsi, il est souvent nécessaire de collecter un volume très important de valeurs afin de consolider un échantillon suffisamment représentatif du comportement global du système. Des conclusions concernant le comportement du système analysé sont alors souvent difficiles à établir avec certitude. Aussi, le plus souvent, seules des potentialités et des approximations de la dynamique et des performances peuvent être dégagées.

Les méthodes analytiques sont basées sur des modèles mathématiques permettant l'obtention de résultats par calcul. La dynamique du système étudié est alors représentée sous formes d'équations. La précision des résultats dépend directement des détails pris en compte par le modèle et de son niveau de corrélation avec le système réel analysé. Les méthodes analytiques autorisent dans certains cas une évaluation fine des performances d'un système. Toutefois, les modèles sur lesquels elles reposent sont généralement établis à partir d'hypothèses simplificatrices ne permettant le plus souvent d'aboutir qu'à des résultats approximatifs. Dans la plupart des cas, la modélisation analytique reste difficile-

ment applicable aux systèmes complexes.

L'analyse de performances à partir d'opérations de simulation représente une alternative courante aux méthodes analytiques. La technique de simulation permet d'analyser des systèmes de taille industrielle mais impose cependant que des hypothèses simplificatrices soient retenues. La dynamique des systèmes étudiés est exprimée par du code algorithmique. La méthode par simulation nécessite généralement beaucoup de temps pour exécuter le modèle selon différentes configurations. Elle requiert ensuite d'isoler les critères cruciaux à l'étude du système à partir d'un important volume de données recueillies avant de mener un travail d'analyse statistique. Bien que la simulation ne fournisse que rarement des résultats tout à fait exacts, il est toujours possible d'évaluer leur degré de précision. Les systèmes de production sont généralement étudiés au moyen de la méthode par simulation.

Le tableau 4.1 présente une synthèse comparative des trois méthodes d'analyse de performances en se basant sur des critères élémentaires.

Critère	Mesures	Méthodes analytiques	Simulation
Durée	Variable	Courte	Moyenne
Outils	Instrumentation	Modèles mathématiques	Langages informatiques
Paramétrage	Difficile	Simple	Moyen
Exactitude	Variable	Variable	Moyenne
Coût	Élevé	Faible	Moyen

TAB. 4.1: *Comparaison des méthodes d'analyse de performances*

La performance d'un système de production est en grande partie établie par ses propriétés quantitatives qui lui assurent une rentabilité économique. Cependant, le comportement d'un système réel est souvent difficile à prédire avec suffisamment de précision. En effet, les systèmes de production deviennent de plus en plus complexes et imposent de prendre en compte un volume important de détails. Il est donc largement fait appel aux techniques de modélisation et de simulation pour estimer a priori le niveau de performances d'un équipement manufacturier en cours de conception. Dans ce contexte, l'évaluation des performances repose souvent sur l'exploitation d'un modèle à l'aide de calculateurs. La modélisation et la simulation permettent alors de tester des hypothèses afin de résoudre les problèmes constatés ou d'atteindre les objectifs visés en raisonnant sur l'exécution du modèle et en analysant les résultats collectés.

Nos travaux s'appliquent à l'analyse de performances de lignes d'assemblage automatisées pouvant être constituées de plusieurs dizaines de machines de production interconnectées et sollicitant régulièrement un contrôleur partagé. Le procédé est alors complexe et son étude par simulation apparaît comme la méthode la plus adaptée compte tenu de la dimension industrielle du système considéré. De plus, la littérature montre que la méthode par simulation est largement employée lorsque l'étude concerne l'évaluation de règles de tri ou, plus largement, lorsqu'elle s'inscrit dans le cadre de la gestion à court terme de systèmes de production [Erickson et al., 1987] [Drake and Smith, 1996] [Allaoui and Artiba, 2004] [Kuhl and Laubisch, 2004] [Persson et al., 2006].

4.2.3 Détail des phases de modélisation et de simulation

La simulation consiste à exécuter et étudier numériquement un modèle dont les caractéristiques sont ensuite évaluées à partir de données collectées [Banks, 1998]. Elle fait aujourd'hui largement appel à des calculateurs informatiques [Bel and Dubois, 1985]. La définition de la simulation est le plus souvent étendue aux notions conjuguées de production et d'exécution d'un modèle [Castagna, 2004] :

Définition : La simulation consiste à construire un modèle d'un système réel et à conduire des expérimentations sur ce modèle afin de comprendre le comportement de ce système et d'en améliorer les performances.

La modélisation est une phase d'identification et de transcription d'un système réel par une théorie, un langage symbolique ou une représentation logico-mathématique dans le but d'en évaluer le comportement et les performances [Monsef, 1996]. Cette évaluation s'effectue selon un contexte et une problématique donnés [Cernault, 1988]. La modélisation s'appuie ainsi sur une mise en œuvre d'un ensemble de techniques afin de disposer d'une expression du système étudié.

Un modèle est une structure simplifiée d'un système réel qui n'intègre que les caractéristiques estimées nécessaires à l'analyse du système étudié. Il s'agit ainsi d'une approximation plus ou moins précise d'un système réel permettant de l'appréhender de manière plus simple en fonction des objectifs visés. Un modèle est alors une représentation subjective d'un système exprimée par des mathématiques, des symboles ou des mots [Calvez, 1990].

4.3 Choix d'un formalisme de modélisation

Contrairement aux systèmes continus, les systèmes à événements discrets ne peuvent pas être décrits à l'aide d'équations différentielles en raison de la nature des phénomènes pris en compte. Il existe dans la littérature plusieurs formalismes permettant d'exprimer le modèle d'un système à événements discrets. Le choix du formalisme dépend de différents critères, mais le plus souvent, il est lié à la nature du système étudié.

4.3.1 Principaux outils pour l'étude de systèmes à événements discrets

Les systèmes à événements discrets sont généralement représentés par des modèles basés sur les notions d'état et de transition. Les outils les plus répandus sont les automates d'états finis, les chaînes de Markov, le Grafcet, les réseaux de Petri et l'algèbre des dioïdes.

Les automates d'états finis sont surtout utilisés pour représenter des systèmes déterministes simples [Hopcroft and Ullman, 1979] alors que les chaînes de Markov s'appliquent plus spécifiquement aux systèmes stochastiques [Haverkort, 2000]. Le Grafcet est un formalisme plus particulièrement utilisé par la communauté des automaticiens [Zaytoon et al., 1997] [Le Parc et al., 1999] [Ndjad and Zaytoon, 1999].

Les réseaux de Petri ont été introduits en 1962 en vue de modéliser et d'analyser des systèmes de communication [Petri, 1962]. Ils sont souvent utilisés pour la modélisation de systèmes dirigés par les événements présentant des caractéristiques de concurrence, d'asynchronisme, d'interblocage et de conflit. Ils sont graphiquement représentés et reposent sur un formalisme mathématique rigoureux particulièrement adapté à l'étude des systèmes à événements discrets [Brams, 1983] [Murata, 1989].

L'algèbre des dioïdes reposent sur une sous-classe des réseaux de Petri prenant en compte la temporisation des événements [Baccelli et al., 1992] [Gaubert, 1992]. Il permet la modélisation d'actions coopérantes se transmettant des résultats mais il n'offre pas la possibilité d'exprimer des choix ou des conflits d'allocation de ressources. L'algèbre des dioïdes se caractérise alors par une limitation à certaines classes de systèmes à événements discrets qui ne considèrent que des phénomènes de synchronisation excluant ainsi les contraintes de concurrence.

4.3.2 Intérêts des réseaux de Petri pour l'analyse de performances des systèmes de production

L'intérêt des réseaux de Petri pour la conception et l'évaluation des systèmes de production a largement été souligné [Chappe and Bourjault, 1988] [Proth et al., 1993] [Desrochers and Al-Jaar, 1995] [Silva et al., 1998] [Zhou and Venkatesh, 1999]. Les résultats obtenus permettent une analyse comportementale et structurelle nécessaire à la validation fonctionnelle du système étudié [DiCesare et al., 1993] [Proth and Xie, 1994] [Tsinarakis et al., 2005].

Les réseaux de Petri s'appuient sur une description des relations existantes entre des conditions et des événements. Une abondante littérature [Barad, 1998] [Heiner, 1997] [Lamch, 2002] décrit les diverses méthodes d'analyse permettant, à partir d'un modèle, d'étudier les propriétés intéressantes d'un système. L'analyse d'un réseau de Petri et la vérification de ses propriétés peuvent révéler des caractéristiques importantes du système concernant ses aspects structurel et comportemental. Toutefois, et bien que les réseaux de Petri permettent d'abord de vérifier des propriétés qualitatives, les résultats qu'ils apportent peuvent être utilisés pour analyser quantitativement le système considéré. Ils sont alors couramment employés dans le contexte d'une évaluation de performances de systèmes [Sifakis, 1977] [van der Aalst, 1994a] [Liu, 1995] [Balbo et al., 1995] [Marsan et al., 1996] [Zimmermann et al., 1996] [Zimmermann and Hommel, 1999].

Parmi les extensions apportées aux réseaux de Petri ordinaires et que nous retenons pour notre travail de modélisation, la temporisation est couramment utilisée pour la modélisation de systèmes de production. Les réseaux de Petri temporisés permettent d'intégrer au modèle des informations temporelles destinées à fixer la durée des activités. Leur utilisation permet de rendre compte de la complexité des systèmes étudiés et confère au modèle un aspect dynamique [Zuberek, 1991] [Alla and David, 1992]. Le comportement des systèmes informatisés de pilotage rencontrés dans les domaines de la productique et de la robotique est très souvent basé sur des mécanismes de temporisation permettant de définir la durée de déroulement de certaines activités. Les extensions temporelles des réseaux de Petri sont des modèles bien adaptés à l'étude formelle de ces mécanismes [Juanole et al., 2000]. Les réseaux de Petri temporisés sont alors largement utilisés pour l'évaluation de performances de systèmes [Ramchandani, 1973] [Razouk and Phelps, 1984] [Holliday and Vernon, 1985] [Gaujal et al., 1993].

Bien que les réseaux de Petri ordinaires et temporisés soient très utilisés pour l'évaluation des performances des systèmes en général et des procédés industriels de manufacture en particulier, ils sont souvent critiqués en raison de la taille qu'ils peuvent prendre lorsque le nombre de produits ou de machines impliqués est important. L'extension colorée des réseaux de Petri permet de contourner ce problème [Jensen, 1992] [Jensen, 1994] [Jensen, 1997]. Elle vise la réduction de la taille du modèle pour aboutir à une représentation plus compacte des systèmes affichant des composantes ayant des comportements identiques. L'extension colorée est fréquemment utilisée pour l'étude des systèmes de production [Martinez et al., 1987] [Viswanadham and Narahari, 1987] [Kasturia et al., 1988] [Colombo et al., 1994] [Moncelet et al., 1998]. Elle convient donc à la modélisation de lignes d'assemblage déclinées en une série de machines dont la dynamique générale est homogène.

L'utilisation conjointe des extensions temporisée et colorée des réseaux de Petri représentent un moyen bien adapté à la modélisation de systèmes complexes de taille importante incluant des contraintes de temps [van der Aalst, 1992] [van der Aalst, 1994b]. Notre objectif étant de bâtir un modèle destiné à l'évaluation de lignes d'assemblage de taille industrielle, le formalisme des réseaux de Petri colorés temporisés convient particulièrement à notre travail de modélisation.

4.4 Description des éléments du modèle

Les principaux éléments du modèle sont l'ensemble des machines de production et le contrôleur central. Afin faciliter sa compréhension, nous présentons d'abord le modèle en plusieurs parties afin d'exposer la manière de représenter le système étudié. Seule l'extension temporisée est alors utilisée. Ensuite, nous utilisons la coloration de jetons pour synthétiser et compacter le modèle afin d'en réduire la taille avant de fusionner les différentes parties.

Par convention, les noms de places et de transitions sont respectivement préfixés par $P_$ et $T_$. Conformément aux notations préalablement retenues au chapitre 3, les éléments du réseau de Petri relatifs aux opérations affectées à une machine sont nommés par la lettre O . Ils sont indicés en fonction de la machine concernée et de leur ordre d'exécution dans le cycle opératoire. Ainsi, la présence d'un jeton dans la place $P_O_{i,j}$ exprime l'exécution de l'opération $O_{i,j}$ sur la machine M_i .

Les opérations distantes, correspondant au traitement d'un message de requête par le contrôleur, sont différenciées des opérations mécaniques exécutées localement par le caractère astérisque placé en exposant.

Notre modèle est principalement décrit à l'aide de l'extension temporisée des réseaux de Petri. De manière générale, la temporisation nous permet de modéliser la durée des activités du système de production et plus particulièrement celle des différentes opérations. Nous associons les temporisations aux transitions pour exprimer les durées opératoires. Notre modèle est alors de type T-temporisé. La valeur d'une temporisation est indiquée par le caractère arobase. La figure 4.1 montre la manière avec laquelle nous modélisons la durée d'exécution d'une action.

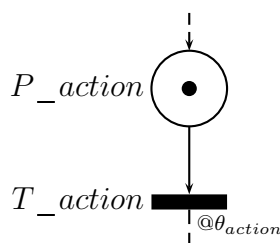


FIG. 4.1: Modélisation de la durée d'exécution d'une action

Le principe repose sur une temporisation du franchissement de la transition T_action retardant ainsi la consommation du jeton situé dans la place P_action . La durée de temporisation est fixée par θ_{action} .

4.4.1 Activité d'une machine de production

Les machines de production reposent toutes sur le même principe de modélisation. Elles sont essentiellement représentées par leur cycle opératoire décliné sous la forme d'un enchaînement d'opérations locales et distantes. Un premier jeton symbolise l'action en cours qu'il s'agisse d'une opération, de la prise en compte d'une pièce ou de son relâchement après traitement. Aucune exécution simultanée de plusieurs opérations ne peut donc se produire. Un second jeton marque la disponibilité de la machine de production modélisée. La façon de modéliser une machine M_i est présentée par la figure 4.2. La taille de cette partie du modèle est directement liée au nombre d'opérations affectées à la machine M_i .

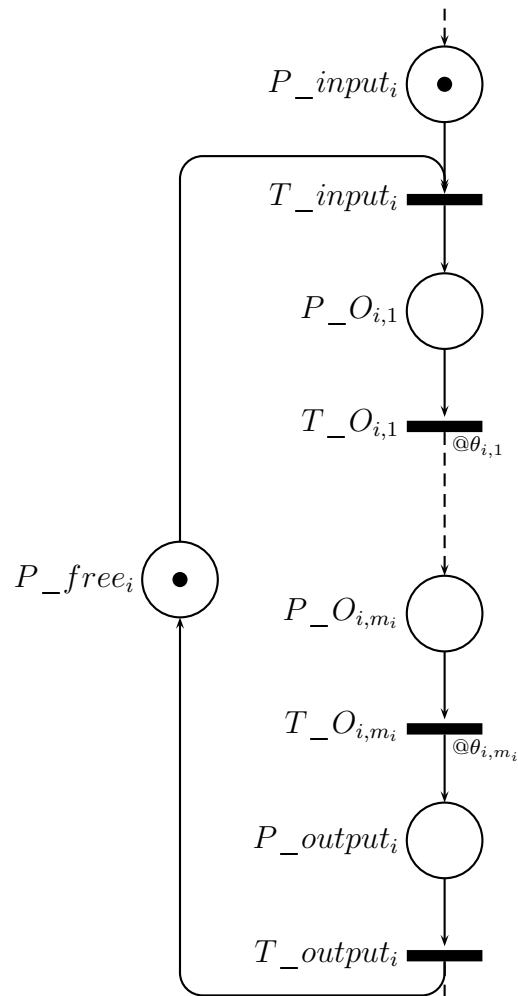


FIG. 4.2: Modélisation d'une machine de production

La place, nommée P_free_i , impose un traitement unitaire des pièces par un accès en exclusion mutuelle à la machine M_i . Les places et les transitions représentant l'entrée et la sortie des pièces de la machine sont désignées par P_input_i , T_input_i , P_output_i et T_output_i . La présence d'un jeton dans la place P_input_i symbolise la disponibilité d'une pièce à l'entrée de la machine et constitue l'une des deux conditions nécessaires au démarrage de son cycle opératoire. La seconde condition est remplie par l'inactivité préalable de la machine exprimée par la présence d'un jeton dans la place P_free_i . Le franchissement de la transition T_input_i marque alors le démarrage du cycle opératoire de la machine.

Les places P_input_i et P_output_i ne sont pas contraintes par une durée définie de rétention des jetons. En effet, la place P_input est destinée à modéliser l'attente des pièces avant le démarrage de leur traitement en cas d'indisponibilité de la machine. La place P_output est utilisée pour exprimer la rétention des pièces sur la machine au terme de leur traitement. Cette situation est provoquée par la saturation de la zone d'encours entre les machines successives M_i et M_{i+1} .

La présence d'un jeton dans les places $P_O_{i,1}$ à P_O_{i,m_i} , représentant l'exécution des opérations affectées à la machine M_i , est temporisée en fonction de la durée chaque opération modélisée. Le délai de rétention d'un jeton dans une place liée à une opération est indirectement paramétré par la temporisation assignée à la transition associée.

4.4.2 Enchaînement des machines de production

La modélisation de l'interconnexion de deux machines consiste à représenter la phase d'acheminement des pièces en tenant compte de la capacité d'encours et la durée de transfert. La figure 4.3 illustre la configuration adoptée au niveau du modèle.

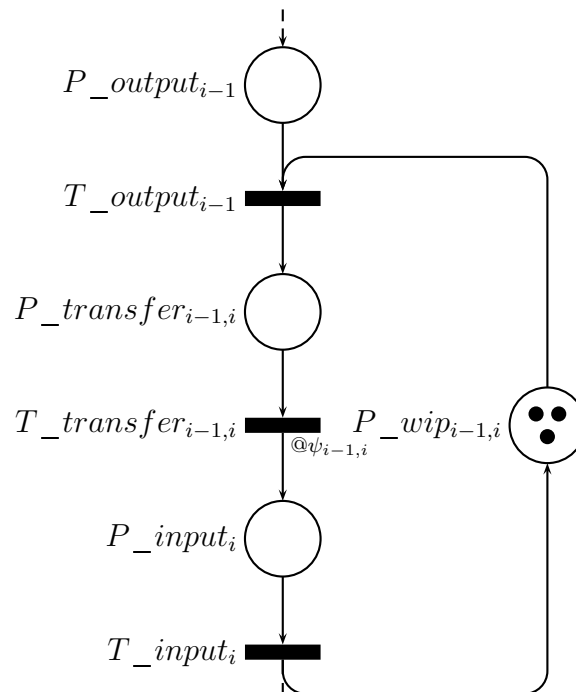


FIG. 4.3: Modélisation de l'interconnexion de deux machines

La capacité d'encours est modélisée par une place notée $P_wip_{i-1,i}$ et initialement marquée selon le nombre maximal de pièces pouvant être présentes entre deux machines de production successives M_{i-1} et M_i . Une pièce est placée en situation d'encours par le franchissement de la transition T_output_{i-1} dès lors qu'elle est libérée par la machine amont M_{i-1} au terme de l'exécution du cycle opératoire et que le nombre de jetons dans la place $P_wip_{i-1,i}$ n'est pas épuisé. Elle y est maintenue jusqu'à sa prise en charge par la machine aval M_i illustrée par le franchissement de la transition T_input_i . L'entrée d'une pièce dans la machine aval M_i provoque l'incrément du nombre de jetons présents dans la place $P_wip_{i-1,i}$ libérant ainsi une position d'encours.

La phase de transfert est représentée par une place unique notée $P_transfer_{i-1,i}$. Le franchissement de la transition $T_transfer_{i-1,i}$, temporisée par la durée $\psi_{i-1,i}$, marque le terme de l'acheminement de la pièce concernée et son arrivée à l'entrée de la machine M_i .

4.4.3 Traitement de messages de requête par le contrôleur

Le contrôleur central est une ressource partagée par l'ensemble des machines de production. Son expression dans le réseau de Petri se limite à une unique place, notée P_idle , identiquement reliée à chacune des parties du modèle représentant une machine de production. Le seul jeton présent dans la place P_idle à l'initialisation du modèle implique l'unicité du traitement de message de requête par le contrôleur. La présence du jeton dans la place P_idle marque la disponibilité du contrôleur. En cas de sollicitations concurrentes du contrôleur, la sélection de la machine de production à servir, et donc du message à traiter, s'effectue selon une règle de tri codée au niveau du logiciel de simulation et conditionnant l'exécution du modèle.

Le modèle doit pouvoir exprimer les cas distincts où le traitement des messages est interruptible et ininterruptible. Nous modélisons alors de deux manières l'activité du contrôleur pour les besoins de notre étude selon qu'elle s'effectue dans un contexte d'interaction préemptive ou non-préemptive.

4.4.3.1 Cas non-préemptif et préemptif

Dans le cas où le traitement des messages ne peut être préempté, la sélection du prochain message s'effectue uniquement lorsque la disponibilité du contrôleur redevient

effective au terme du traitement complet du message en cours. La figure 4.4 montre le modèle de base de l'interaction entre une machine M_i de production et le contrôleur central à travers un exemple d'exécution de cinq opérations. Nous avons donc $m_i = 5$.

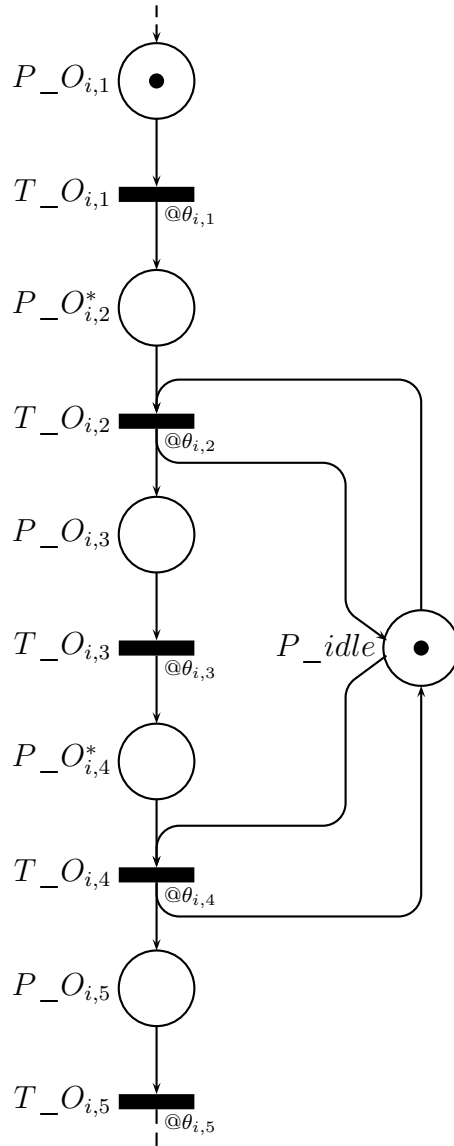


FIG. 4.4: Modélisation de l'interaction non-préemptible du contrôleur avec une machine

Les deuxième et quatrième opérations, respectivement représentées par les places $P_{O_{i,2}^*}$ et $P_{O_{i,4}^*}$ dont la temporisation est fixée par $\theta_{i,2}$ et $\theta_{i,4}$, correspondent au traitement d'un message de requête par le contrôleur. Le franchissement des transi-

tions $T_{_O_{i,2}}$ et $T_{_O_{i,4}}$ modélise alors la fin de traitement par le contrôleur du message de requête lié à l'opération distante et l'envoi du message de réponse à la machine M_i .

La manière de modéliser l'activité interrompible du contrôleur est une variante du cas non-préemptif. Elle est présentée par la figure 4.5 basée sur l'exemple précédent.

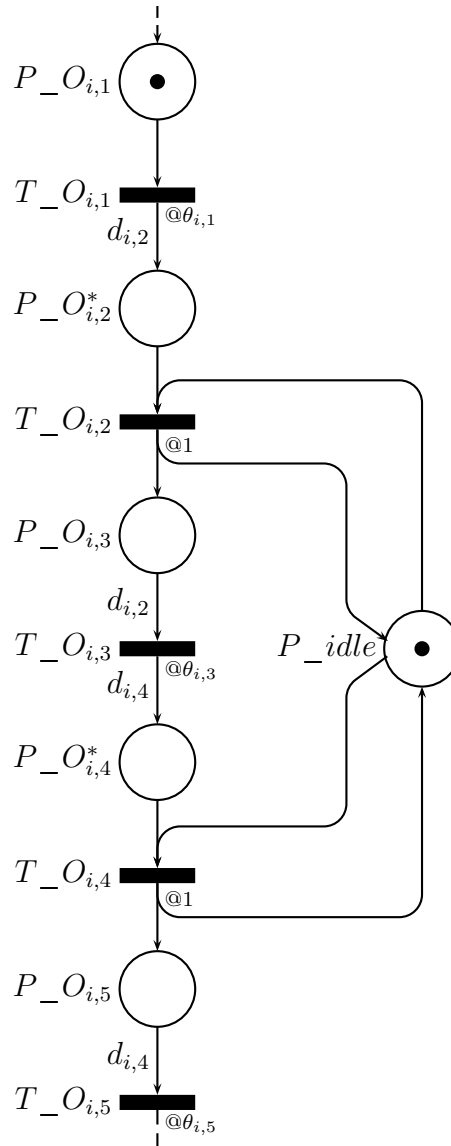


FIG. 4.5: Modélisation de l'interaction préemptible du contrôleur avec une machine

La modélisation de l'interruption des opérations relatives à un traitement de message de requête par le contrôleur central s'appuie sur une décomposition en quantums de

temps de la durée nécessaire à leur exécution. La durée de traitement par le contrôleur d'un message de requête correspondant à une opération $O_{i,j}^*$ affectée à la machine M_i , que nous notons $d_{i,j}$, est renseignée au niveau du modèle par une valuation des arcs reliant la transition $T_{O_{i,j-1}}$ à la place $P_{O_{i,j}^*}$ et la place $P_{O_{i,j+1}}$ à la transition $T_{O_{i,j+1}}$. Dans le cas présenté, les places $P_{O_{i,2}^*}$ et $P_{O_{i,4}^*}$ sont donc initialisées lors du franchissement des transitions $T_{O_{i,1}}$ et $T_{O_{i,3}}$ avec un nombre de jetons correspondant à la durée des opérations qu'elles symbolisent.

Le principe de modélisation repose alors sur un nombre de $d_{i,j}$ rotations du jeton initialement présent dans la place P_{idle} pour fixer la durée de traitement du message de requête par le contrôleur central. Il s'agit ainsi de valider le passage d'une opération à la suivante en consommant unitairement les $d_{i,j}$ jetons insérés dans la place $P_{O_{i,j}^*}$ concernée.

Au terme de chaque quantum de temps, le jeton est replacé dans la place P_{idle} afin de permettre son attribution éventuelle à une autre machine. Dans le cas où le quantum de temps écoulé n'est pas le dernier, l'opération distante en cours d'exécution est suspendue gelant ainsi le déroulement du cycle opératoire de la machine. La poursuite de son exécution est conditionnée par la disponibilité et l'attribution du jeton présent dans la place P_{idle} .

4.4.3.2 Sélection de la machine à servir

Le contrôleur central est une ressource du système dont l'activité est partagée par l'ensemble des machines de production. Le choix de la machine à servir par l'implémentation d'une règle de tri au niveau du logiciel de simulation représente donc un aspect important de la modélisation. Il consiste à arbitrer la consommation du jeton présent dans la place P_{idle} .

Un exemple de concurrence d'accès au contrôleur par deux machines de production implantées successivement sur le système M_i et M_{i+1} est montré par la figure 4.6. Il reprend les exemples proposés aux figures 4.4 et 4.5 pour lesquels nous retenons pour chaque machine une séquence opératoire identique constituée de cinq opérations. Le franchissement des transitions $T_{O_{i,2}}$, $T_{O_{i+1,2}}$, $T_{O_{i,4}}$ et $T_{O_{i+1,4}}$ respecte une contrainte d'exclusion mutuelle imposant à une des deux machines à suspendre son cycle opératoire pendant que l'autre est servie par le contrôleur.

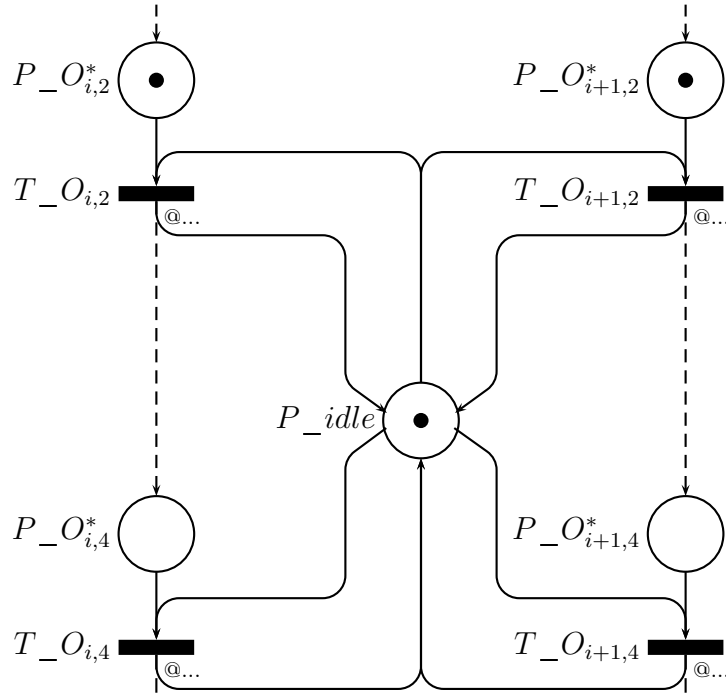


FIG. 4.6: Modélisation de la concurrence d'accès au contrôleur

La temporisation des transitions $T_{O_{i,2}}$, $T_{O_{i+1,2}}$, $T_{O_{i,4}}$ et $T_{O_{i+1,4}}$ dépend du mode d'interaction préemptif ou non-préemptif des machines avec le contrôleur.

4.4.4 Construction du modèle coloré temporisé

La coloration des jetons est utilisée pour distinguer chacune des machines de production implantées sur le système modélisé, les différentes opérations et dans certains cas la durée d'exécution résiduelle des opérations. Les couleurs associées sont alors respectivement notées sur le modèle en fonction des indices i , j et d .

La modélisation haut niveau, essentiellement basée sur la temporisation des transitions et la coloration des jetons, implique la prise en compte de certaines notations spécifiques que nous retenons. Ainsi, les couleurs associées à un jeton évoluant d'une place vers une transition ou d'une transition vers une place sont exprimées entre parenthèses. Lorsqu'une couleur fait l'objet d'une incrémentation ou d'une décrémentation, elle est respectivement complétée par une flèche montante ou descendante. Enfin, les conditions particulières nécessaires au franchissement d'une transition sont notées en-

tre crochets. La notion de *garde* est le plus souvent employée pour évoquer ce type de contrainte.

La modélisation haut niveau d'une machine de production M_i est présentée par la figure 4.7. Les éléments nécessaires à la représentation de l'ensemble \mathcal{O}_i des opérations affectées à la machine M_i se limitent à l'expression de la place $P_{-O_{i,j}}$ et des transitions $T_{-O_{i,j}}$ et $T_{-O_{i,m_i}}$ temporisées en fonction de la durée $\theta_{i,j}$ de l'opération en cours d'exécution. Les gardes $[j < m_i]$ et $[j = m_i]$ servent à distinguer la dernière opération O_{i,m_i} de la gamme affectée à la machine de production M_i des précédentes. Son terme doit en effet être différemment traité pour conditionner la suspension d'activité de la machine afin de la rendre disponible pour l'exécution du travail suivant par l'insertion d'un jeton de couleur i dans la place P_{-free_i} . Les places P_{-input_i} et P_{-free_i} sont initialisées avec un jeton de couleur i .

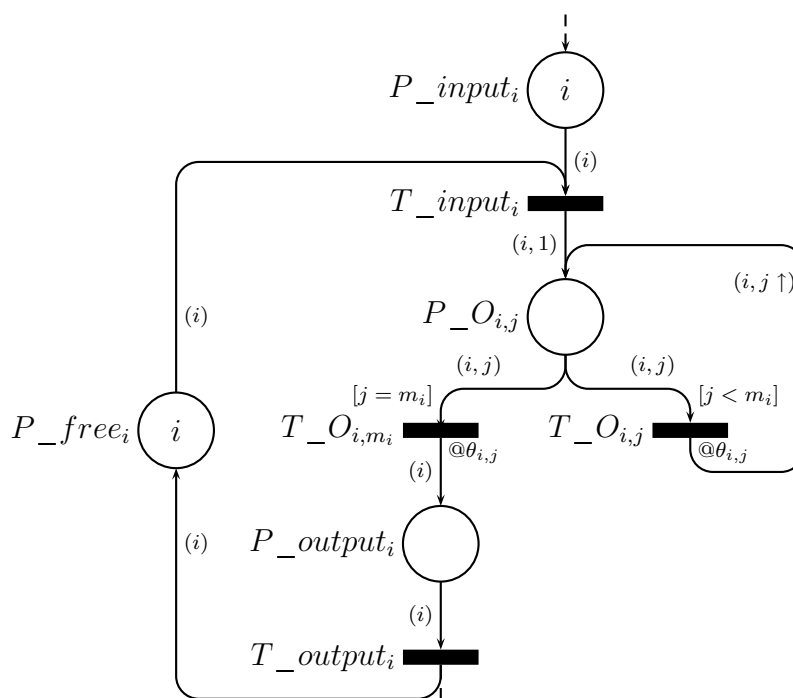


FIG. 4.7: Modélisation haut niveau d'une machine de production

La modélisation haut niveau de l'interconnexion de deux machines M_{i-1} et M_i implantées successivement sur le système considéré est présentée par la figure 4.8. La coloration permet de n'utiliser que quatre places et trois transitions pour l'ensemble

du système. La durée $\psi_{i-1,i}$ du transfert des pièces entre les deux machines est fixée par la temporisation de la transition $T_transfer_{i-1,i}$. La capacité d'encours entre les deux machines est définie en initialisant la place $P_wip_{i-1,i}$ avec un nombre $\varpi_{i-1,i}$ de jetons de couleur i . Le passage de la pièce transférée de la machine M_{i-1} vers la machine M_i est symbolisé par l'incrément de la couleur i du jeton envoyé dans la place P_input_i . La prise en charge par la machine M_i d'une pièce en attente à son entrée provoque l'insertion d'un jeton de couleur $i-1$ dans la place $P_wip_{i-1,i}$ marquant ainsi la décrémentation du stock de pièces en encours. De manière complémentaire, la mise en encours d'une pièce par la machine de production M_i , symbolisée par le franchissement de la transition T_output_i , implique qu'un jeton de couleur i soit disponible dans la place $P_wip_{i-1,i}$. Le transfert de pièces à travers le système représenté n'impliquant que la seule distinction des machines, une unique couleur de jetons suffit à la construction de cette partie du modèle.

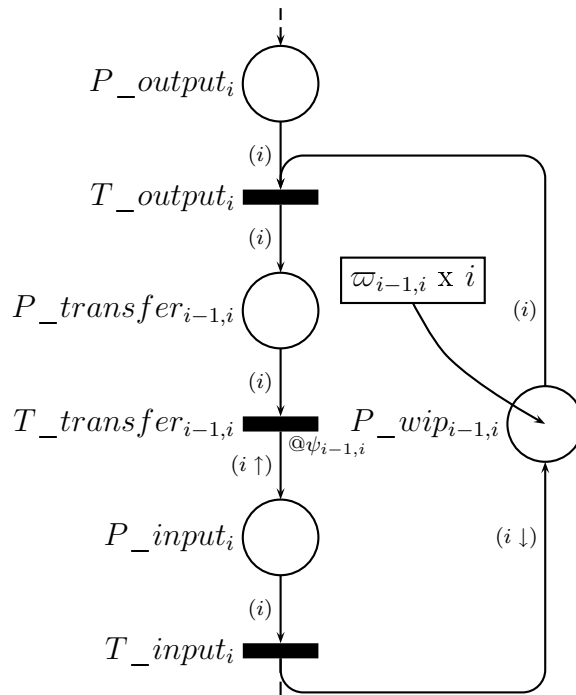


FIG. 4.8: Modélisation haut niveau de l'interconnexion de deux machines

Concernant l'interaction du contrôleur central avec les machines de production, la modélisation du cas non-préemptif impose de prendre en compte quatre types d'opéra-

tions correspondant chacun à un routage particulier des jetons. Tout d'abord, les opérations peuvent être exécutées localement par la machine ou confiées au contrôleur. Il est ensuite nécessaire de considérer différemment la fin d'exécution de la dernière opération affectée à la machine M_i de celle des précédentes par les gardes $[j < m_i]$ et $[j = m_i]$. La Nous pouvons alors isoler tous les types d'opérations et traiter spécifiquement chaque cas de fin d'exécution. Le franchissement de la transition $T_{-}O_{i,j}$, associée à la garde $[j < m_i]$ marque le terme d'une opération locale autre que la dernière assignée à la machine M_i . Celui de la transition $T_{-}O_{i,m_i}$, associée à la garde $[j = m_i]$, est conditionné par la fin d'exécution d'une opération locale correspondant à la dernière assignée à la machine M_i . Les transitions complémentaires $T_{-}O_{i,j}^*$ et $T_{-}O_{i,m_i}^*$ s'appliquent de manière identique aux cas des opérations distantes exécutées par le contrôleur. Le tableau 4.2 présente un récapitulatif détaillé des quatre transitions nécessaires à la modélisation de l'interaction non-préemptible du contrôleur central avec les machines de production.

Transition	Condition	Description
$T_{-}O_{i,j}$	$j < m_i$	Fin d'exécution locale d'une opération $O_{i,j}$ autre que la dernière affectée à la machine M_i
$T_{-}O_{i,m_i}$	$j = m_i$	Fin d'exécution locale de la dernière opération O_{i,m_i} affectée à la machine M_i
$T_{-}O_{i,j}^*$	$j < m_i$	Fin d'exécution distante d'une opération $O_{i,j}$ autre que la dernière affectée à la machine M_i
$T_{-}O_{i,m_i}^*$	$j = m_i$	Fin d'exécution distante de la dernière opération O_{i,m_i} affectée à la machine M_i

TAB. 4.2: *Détail des transitions liées à l'interaction non-préemptible du contrôleur*

La figure 4.9 montre la manière avec laquelle nous modélisons l'interaction non-préemptible du contrôleur avec une machine M_i . La coloration des jetons permet de ne considérer qu'une unique place pour représenter l'ensemble des opérations de la gamme affectées à la machine. La place concernée, notée $P_{-}O_{i,j}$, est temporisée par la transition $T_{-}O_{i,j}$ en fonction de la durée $\theta_{i,j}$ d'exécution de l'opération $O_{i,j}$. La distinction des machines et des opérations s'appuie sur une utilisation conjointe de deux couleurs affectées aux jetons dont la circulation symbolise l'activité du système modélisé. La consommation de l'unique jeton disponible dans la place $P_{-}idle$ est effectuée en fonction d'une règle de tri définie au niveau du logiciel de simulation utilisé. Elle reflète

l'attribution de la ressource partagée que représente le contrôleur à la machine de production évaluée comme étant la plus prioritaire lorsque le choix est arbitrée au terme de l'exécution d'une opération distante.

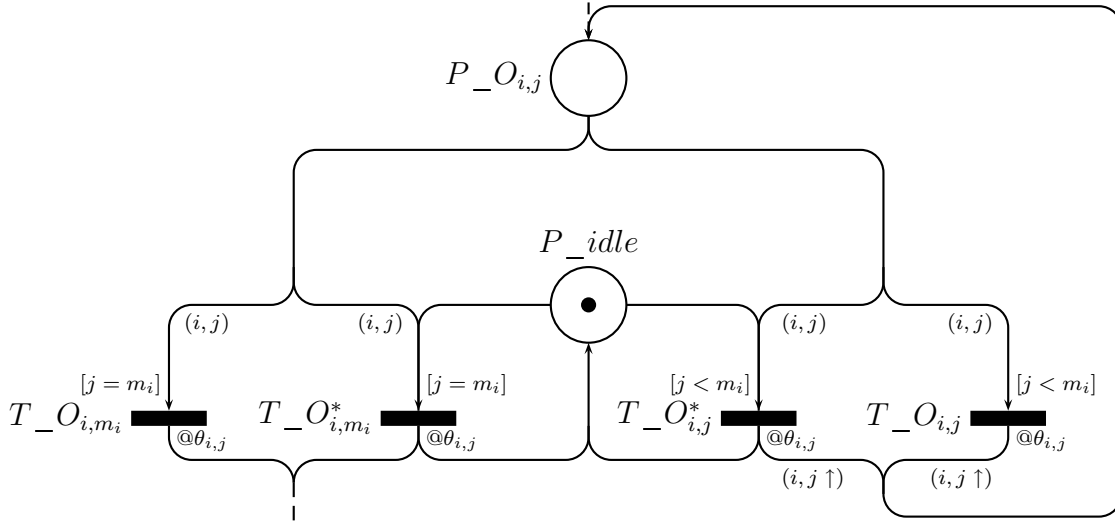


FIG. 4.9: Modélisation haut niveau de l'interaction du contrôleur avec une machine

La modélisation présentée par la figure 4.9 se limite à l'expression d'une interaction non-préemptible entre les machines de production et le contrôleur central. Il convient alors d'étendre le modèle afin d'exprimer le mécanisme de préemption. L'opération distante à exécuter, et donc la machine de production à servir, doivent faire l'objet d'une évaluation permanente au cours de l'activité du système. La méthode de segmentation des opérations en quantum de temps permet, à l'aide de la règle de tri implémentée, une sélection de la machine à servir au terme de l'exécution de chaque quantum de temps des opérations distantes confiées au contrôleur. Les opérations étant segmentées, la modélisation doit prendre en compte et gérer la durée d'exécution résiduelle d , pour l'opération en cours sur chacune des machines du système. De manière similaire au cas non-préemptif, quatre principales transitions sont alors nécessaires pour tester la fin d'exécution d'un quantum de temps d'opération. Leur franchissement est conditionné par une garde, $[d > 0]$ ou $[d = 0]$, destinée à évaluer la durée résiduelle de son exécution. La transition $T_{O_{i,j,d}}$, associée à la garde $[d > 0]$, est franchie après l'exécution d'un quantum de temps d'une opération locale en cours alors que la transition $T_{O_{i,j,0}}$, associée à la garde $[d = 0]$, s'applique identiquement au traitement d'une opération

locale qui s'achève. De manière complémentaire, les transitions $T_{-O_{i,j,d}^*}$ et $T_{-O_{i,j,0}^*}$ correspondent au cas d'une opération distante exécutée par le contrôleur. Deux transitions supplémentaires, notées $T_{-O_{i,j}}$ et $T_{-O_{i,m_i}}$, sont utilisées pour gérer les cas de fin d'exécution local ou distante d'une opération. Le tableau 4.3 résume la description des six transitions utilisées pour modéliser l'interaction préemptible du contrôleur central avec les machines de production.

Transition	Condition	Description
$T_{-O_{i,j,d}}$	$d > 0$	Fin d'exécution locale d'un quantum de temps autre que le dernier d'une opération $O_{i,j}$ affectée à la machine M_i
$T_{-O_{i,j,d}^*}$	$d > 0$	Fin d'exécution distante d'un quantum de temps autre que le dernier d'une opération $O_{i,j}$ affectée à la machine M_i
$T_{-O_{i,j,0}}$	$d = 0$	Fin d'exécution locale du dernier quantum de temps d'une opération $O_{i,j}$ affectée à la machine M_i
$T_{-O_{i,j,0}^*}$	$d = 0$	Fin d'exécution distante du dernier quantum de temps d'une opération $O_{i,j}$ affectée à la machine M_i
$T_{-O_{i,j}}$	$j < m_i$	Fin d'exécution locale ou distante d'une opération $O_{i,j}$ autre que la dernière affectée à la machine M_i
$T_{-O_{i,m_i}}$	$j = m_i$	Fin d'exécution locale ou distante de la dernière opération O_{i,m_i} affectée à la machine M_i

TAB. 4.3: *Détail des transitions liées à l'interaction préemptible du contrôleur*

La manière avec laquelle nous avons modélisé la possibilité de suspendre le traitement d'un message par le contrôleur au profit d'un autre est montrée par la figure 4.10. Cette partie du modèle est construite de façon à assurer un traitement itératif des quantums de temps d'une opération jusqu'à son terme. Aussi, la présence du jeton dans la place $P_{-O_{i,j}}$ est fixée à un quantum de temps par la temporisation des transitions $T_{-O_{i,j,d}}$, $T_{-O_{i,j,d}^*}$, $T_{-O_{i,j,0}}$ et $T_{-O_{i,j,0}^*}$. L'exécution des opérations distantes peut ainsi être préemptée à chaque quantum de temps. Les jetons associés à un quantum de temps sont porteurs de trois couleurs afin de préciser la machine concernée, l'opération en cours et la durée résiduelle d'exécution. La place $P_{-end_{i,j}}$ est uniquement destinée à représenter la fin d'exécution d'une opération après le franchissement des transitions $T_{-O_{i,j,0}}$ et $T_{-O_{i,j,0}^*}$ et le routage des jetons par les transitions $T_{-O_{i,j}}$ et $T_{-O_{i,m_i}}$.

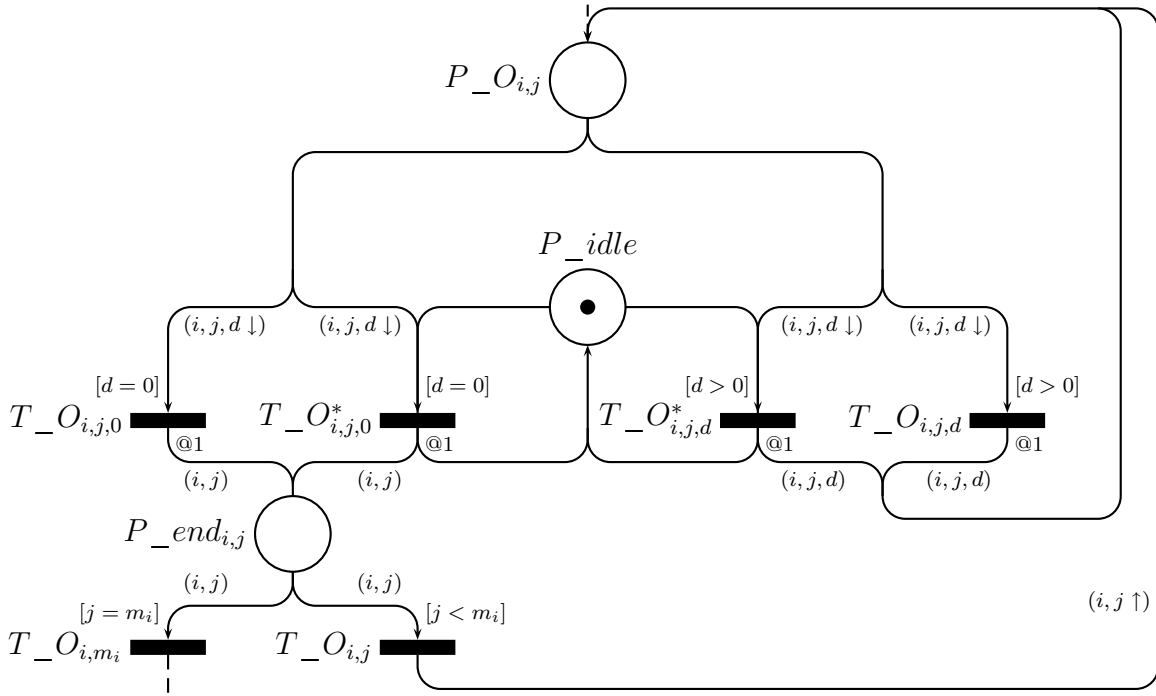


FIG. 4.10: Modélisation haut niveau du mécanisme de préemption

Le modèle complet, dont l'aspect est montré par la figure 4.11, est élaboré en fusionnant les parties présentées par les figures 4.7 à 4.10. La zone délimitée par une ligne de tirets constitue une variante de la structure du modèle. Elle intègre, en fonction de la configuration désirée, la partie décrite par la figure 4.9 ou celle correspondant à la figure 4.10 selon que la règle de tri évaluée durant la phase de simulation soit ou non basée sur un mécanisme de préemption. La transition P_input_1 , associée à la garde $[i = 1]$, permet de gérer la prise en charge d'une pièce par la première machine de manière différente des autres ressources du système. En effet, le franchissement de la transition T_input_i provoque l'incrément du nombre de places d'encours disponibles en amont de la machine M_i par insertion d'un jeton de couleur $i - 1$ dans la place $P_wip_{i-1,i}$. Ce cas de figure ne peut pas s'appliquer à la première machine du système. Aussi, la transition P_input_i est conditionnée par la garde $[i > 1]$ afin d'interdire son franchissement dans le cas où la couleur du jeton symbolise la première machine. De façon similaire, la libération d'une pièce par une machine nécessite également de traiter un cas particulier. Ainsi, lorsque la machine concernée est la dernière ressource du système, pour laquelle

il n'existe pas de limitation d'encours ni de phase de transfert vers la machine suivante, aucune consommation de jeton de couleur k dans la place $P_wip_{i-1,i}$ ne doit être permise au moment de la sortie de la pièce. Le franchissement de la transition T_output_i est donc protégé par la garde $[i < k]$ excluant le cas de la dernière machine notée k uniquement traité par la transition T_output_k et la garde $[i = k]$ qui lui est associée. La place P_stock_k permet de recueillir un jeton de couleur k à chaque franchissement de la transition P_output_k dans le but d'évaluer le volume de pièces produites lors d'une phase de simulation.

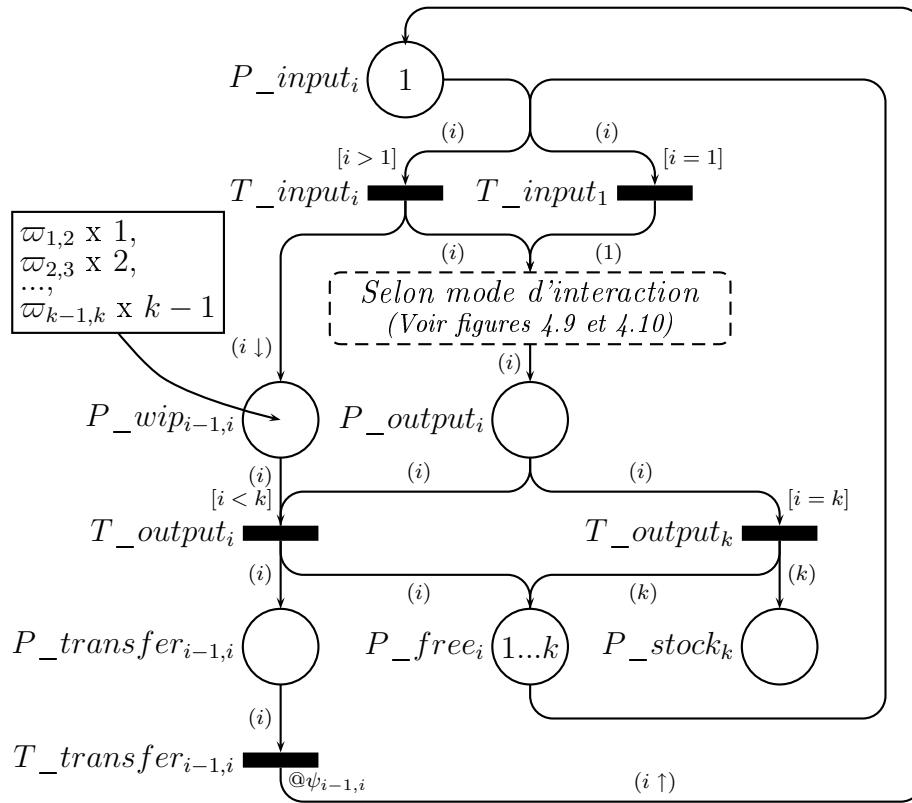


FIG. 4.11: Modélisation haut niveau simplifiée d'un système flow-shop contrôlé

Les trois places P_input_i , $P_wip_{i-1,i}$ et P_free_i nécessitent d'être initialisées préalablement au début de l'exécution du modèle. Un jeton de couleur 1 est alors positionné dans la place P_input_i pour exprimer l'entrée en activité de la première machine de production dès le démarrage de la simulation du système. La place P_free_i est marquée par un jeton de chaque couleur de 1 à k afin de rendre disponible l'ensemble des ma-

chines de production. La capacité d'encours est également initialement fixée pour chacun des $k - 1$ espaces inter-machines par un nombre défini de jetons insérés au niveau de la place $P_{_wip_{i-1,i}}$.

Le modèle que nous proposons aux figures 4.7 à 4.11 est une représentation générique simple d'un réseau de Petri coloré temporisé destinée à faciliter sa compréhension. Toutefois, les opérations de simulation imposent l'utilisation d'un logiciel spécifique et adapté à l'analyse de performances. Nous pouvons plus particulièrement citer les outils CPN Tools² [Ratzer et al., 2003] [Wells, 2006] [Jensen et al., 2007] et TimeNET³ [German et al., 1995] [Zimmermann et al., 2000] [Zimmermann et al., 2001]. Le modèle doit alors le plus souvent être implémenté dans le respect d'un formalisme dédié et de règles de représentation particulières à l'outil mis en œuvre.

4.5 Définition des configurations étudiées

Les travaux présentés dans cette thèse sont inspirés d'un cas réel et notre premier objectif est alors de résoudre la problématique posée par un système de production existant. Le contexte d'analyse de performances doit ainsi être en rapport avec la dimension de procédés de taille industrielle. Par conséquent, les configurations que nous devons en premier lieu étudiées concernent des systèmes incluant jusqu'à trente machines. Le paramétrage des machines est alors initialement défini à partir de mesures relevées sur des lignes d'assemblage existantes. Leur séquence opératoire est également calquée sur l'activité des procédés réels considérés.

La configuration du modèle est déclinée selon différentes variantes. Elle repose sur le nombre de machines de production déclarées, le temps de traitement des messages de requête et la règle de tri définie pour sélectionner la machine à servir par le contrôleur.

Néanmoins, le modèle est construit afin de permettre l'évaluation de systèmes présentant des configurations diverses. Le paramétrage du modèle n'est donc pas restreint à des procédés existants. Il est en effet intéressant de pouvoir étudier des cas plus spécifiques afin de mettre en évidence une propriété particulière des systèmes ou de valider une hypothèse.

²<http://wiki.daimi.au.dk/cpntools/cpntools.wiki>

³<http://pdv.cs.tu-berlin.de/~timenet/>

4.5.1 Paramétrage initial des machines de production

La définition d'un modèle de base repose sur le constat que le taux de dégradation de performances des lignes d'assemblage étudiées est en relation directe avec le nombre de machines implantées. Ce paramètre agissant en effet sur le nombre de messages de requête nécessaires à la production de chaque pièce et donc sur la charge du contrôleur central. Nous avons alors choisi une configuration initiale du modèle qui, lors de la simulation, mette en évidence l'impact du contrôleur sur la cadence de production.

Toutes les machines de la ligne d'assemblage modélisée exécutent une séquence opératoire identique déclinée en cinq opérations :

- pré-assemblage,
- demande de statut,
- assemblage,
- enregistrement de données,
- post-assemblage.

L'opération de pré-assemblage consiste à réceptionner et identifier la pièce à traiter. Elle précède la soumission au contrôleur d'un message de requête porteur d'une demande de statut de la pièce afin d'en autoriser ou d'en interdire l'assemblage selon qu'elle soit déclarée valide ou rebutée par la réponse renvoyée. Dans le cas où la pièce est valide, l'assemblage est autorisé. Il constitue l'opération principale du cycle opératoire. Au terme de l'opération d'assemblage, des données de traçabilité et de suivi de la production sont transmises au contrôleur sous la forme d'un message de requête. Une réponse constituée d'un simple acquittement est alors retournée à la machine restée en attente. Le rebut d'une pièce ne se produit que très rarement sur les systèmes réels à l'origine de nos travaux. Aussi, nous ne considérons pas ce cas particulier dans la définition du modèle dont les seuls effets consistent à écourter la durée du cycle opératoire de la machine et à réduire le nombre de messages de requête soumis au contrôleur.

Sur les systèmes réels, les messages de requête sont acheminés vers le contrôleur par un réseau de terrain en un temps moyen proche de 5 ms. Ce délai de propagation étant très court en comparaison des différents temps opératoires relevés, nous considérons de ne pas l'intégrer au modèle et de l'admettre comme étant nul.

Le tableau 4.4 présente les données de paramétrage communes à toutes les machines de production. Elles correspondent à des valeurs moyennes relevées sur les systèmes

existants à partir desquels est configuré notre modèle. Les temps de transfert de pièces et les capacités d'encours sont affectés de manière identique pour l'ensemble du système modélisé afin de simplifier son paramétrage et son analyse. Ces deux paramètres restent néanmoins proches des valeurs réelles constatées sur les systèmes réels.

Paramètre	Valeur
Durée de transfert	3 000 ms
Capacité d'encours	3 pièces
Délai de pré-assemblage	730 ms
Délai de post-assemblage	60 ms

TAB. 4.4: Paramètres communs à l'ensemble du système

Le paramétrage des machines est différencié par un temps d'assemblage spécifique. Afin de définir des configurations du modèle qui correspondent à un système de production réaliste, nous avons établi les temps d'assemblage à l'aide d'un générateur de nombres aléatoires en bornant les valeurs minimale et maximale par rapport aux machines les plus lentes et les plus rapides implantées sur les systèmes réels. Les temps d'assemblage sont distribués entre 1 100 et 8 500 ms. Le tableau 4.5 montre le temps d'assemblage affecté à chacune des trente machines de production pouvant être déclarées sur le modèle.

Machine	Délai (ms)	Machine	Délai (ms)	Machine	Délai (ms)
1	1 900	11	1 100	21	4 900
2	6 500	12	8 400	22	3 600
3	5 800	13	2 800	23	5 800
4	6 400	14	2 300	24	1 100
5	3 000	15	6 700	25	1 700
6	6 600	16	3 400	26	5 000
7	6 100	17	8 500	27	3 500
8	6 400	18	2 700	28	2 800
9	3 900	19	6 600	29	4 600
10	7 200	20	6 900	30	7 600

TAB. 4.5: Délais d'assemblage pour chacune des machines

Lors de nos relevés sur les systèmes existants, les temps moyens de traitement d'un message de requête par le contrôleur ont été constatés aux environs de 500 ms.

4.5.2 Déclinaisons de configurations pour les systèmes existants

L'étude et l'analyse de performances de systèmes flow-shop contrôlés, tels que ceux que nous considérons dans cette thèse, ne peuvent pas se limiter à la prise en compte d'une seule configuration. Il est donc indispensable de décliner différents paramétrages du modèle construit en se basant sur une variation des principaux critères régissant son comportement.

Le paramétrage initial des machines est conforme à un système flow-shop standard pour lequel la cadence de production τ est fixée par la machine la plus lente. En effet, quelle que soit la configuration retenue, nous relevons que la productivité du système modélisé n'est jamais bornée par une phase de transfert. En regroupant les configurations en fonction de la machine la plus lente nous obtenons six cas de figure donnés par le tableau 4.6.

Nombre de machines	Délai d'inter-sorties de pièces (ms)
1	2 690
2 à 5	7 290
6 à 9	7 390
10 à 11	7 990
12 à 16	9 190
17 à 30	9 290

TAB. 4.6: *Évaluation de la cadence de production*

Les délais d'inter-sorties de pièces du système sont donnés en sommant les délais de pré-assemblage, d'assemblage et de post-assemblage et en considérant des temps de traitement de message de requête nuls. La borne fixée par une phase de transfert est alors systématiquement inférieure à celle établie par la machine la plus lente :

$$\max_{M_i \in [M_2, M_k]} \left\{ \frac{\psi_{i-1,i}}{\varpi_{i-1,i}} \right\} = \frac{3\,000}{3} = 1\,000 \text{ ms} < \max_{M_i \in \mathcal{M}} \{\theta_i\} \quad (4.1)$$

Pour la phase de simulation, et donc d'exécution du modèle à l'aide d'un logiciel, nous avons adopté une stratégie consistant à faire le temps de traitement des messages de requête par le contrôleur central.

En synthèse, l'activité des machines de production déclarées au niveau du modèle

pour la configuration initiale est exprimée sous la forme d'une séquence opératoire unique constituée d'une alternance de cinq opérations locales et distantes. Les deuxième et quatrième opérations correspondent à un traitement de message de requête confié au contrôleur central :

$$\forall M_i \in \mathcal{M} : \mathcal{O}_i = \{O_{i,1}, O_{i,2}^*, O_{i,3}, O_{i,4}^*, O_{i,5}\} \quad (4.2)$$

La durée des opérations $O_{i,1}$, $O_{i,2}^*$, $O_{i,4}^*$ et $O_{i,5}$ est commune pour l'ensemble des machines. Seule l'opération $O_{i,3}$, relative à la phase d'assemblage, présente un temps d'exécution spécifique à chaque machine du système modélisé. La séquence opératoire affectée à chacune des machines doit être paramétrée pour la phase de simulation. L'ordre des opérations, leur durée et leur localisation se définissent au niveau du logiciel de simulation.

4.5.3 Règles de tri évaluées et principal élément de comparaison

De nombreuses études comparatives de règles de tri appliquées à des systèmes de production de type flow-shop ont été menées [Scudder and Hoffmann, 1985] [Sculli, 1987] [Sculli and Tsang, 1990] [Hunsucker and Shan, 1994] [Rajendran and Alicke, 2007]. Elles s'appliquent à la résolution de problèmes d'ordonnancement de l'activité des machines par la sélection de la pièce à traiter.

Le cadre de nos travaux diffère du contexte général par le fait qu'il s'agit d'organiser l'activité d'un contrôleur central partagé par l'ensemble des machines de production. Différentes règles de tri utilisées pour l'ordonnancement des messages doivent alors être évaluées par rapport à une valeur optimale du makespan. Nous considérons a priori que la situation d'optimalité correspond au cas où toutes les opérations sont exécutées localement par les machines de production. La borne sur la cadence de production est alors d'ordre mécanique. Nous établissons donc que l'objectif visé consiste à se rapprocher au plus près du niveau de performances d'un système flow-shop au sein duquel les opérations distantes seraient traitées sans attente, identiquement aux opérations locales, et où la concurrence d'accès au contrôleur serait inexistante.

Les règles de tri courantes que nous avons retenues à des fins de comparaison sont *FIFO*, *LIFO*, *RND*, *SPT* et *LPT*. Nous intégrons également dans la liste des évalua-

tions à mener, deux règles de tri basées sur l'organisation physique des systèmes simulés. Nous les désignons par *Push* et *Pull* pour évoquer respectivement les méthodes utilisées pour sélectionner le message de requête en provenance de la machine la plus en amont et la plus en aval.

4.6 Conclusion

Les aspects de la modélisation des systèmes flow-shop contrôlés ont été abordés dans ce chapitre. Nous y argumentons le choix du formalisme des réseaux de Petri et de leurs extensions colorée et temporisée avant de décrire les éléments qui ont amené à la construction progressive d'un modèle destiné à analyser les performances des procédés étudiés.

Le choix des configurations du modèle que nous retenons est justifié par le contexte de l'étude orienté vers des systèmes flow-shop de taille industrielle. Différentes déclinaisons du paramétrage des machines de production sont également indiquées ainsi que les règles de tri retenues pour ordonnancer l'activité du contrôleur central.

Le travail de simulation à mener à partir du modèle établi est destiné à évaluer comparativement l'impact sur le makespan des règles de tri utilisées au niveau du contrôleur pour sélectionner le message de requête à traiter.

Chapitre 5

Présentation et synthèse des résultats de simulation

Sommaire

5.1	Choix des configurations évaluées et des résultats présentés	115
5.2	Définition et évaluation de la borne mécanique	116
5.3	Analyse de la cadence de production	117
5.3.1	Comparaison des règles de tri	117
5.3.2	Mise en évidence de l'impact du contrôleur	121
5.4	Observation du temps de réponse du contrôleur	122
5.5	Cadence imposée par le contrôleur	123
5.5.1	Formulation d'une borne de contrôle	123
5.5.2	Comparaison des bornes sur la cadence de production	125
5.5.3	Redéfinition de la condition d'optimalité et de la notion de latence	130
5.6	Extension des simulations	131
5.6.1	Choix d'une gamme opératoire et des règles de tri évaluées	131
5.6.2	Résultats d'exécution	132
5.6.3	Étude particulière de la règle de tri <i>FIFO</i>	141
5.6.4	Évaluation d'une configuration bornant la cadence de production par la machine la plus lente	144
5.7	Conclusion	147

L'établissement d'un modèle sous la forme d'un réseau de Petri et son exécution à l'aide d'un logiciel, nous permettent d'obtenir des informations quant au comportement des lignes d'assemblage étudiées. Un paramétrage initial du modèle, reposant sur les caractéristiques de systèmes de production existants a d'abord été mené. Cette étape était destinée à obtenir en premier lieu des données pouvant être comparées à des relevés issus des procédés réels. Elle nous a permis d'aboutir à une qualification préalable du modèle par comparaison des cadences de production constatées. Seule la règle de tri *FIFO* a alors été déclarée puisqu'elle correspond à la discipline utilisée par défaut sur les systèmes existants pour sélectionner le message de requête à traiter par le contrôleur. À partir du modèle qualifié, nous avons ensuite pu mener un travail d'étude des effets induits sur les performances des systèmes par l'ajustement de différents paramètres que sont le nombre de machines de production implantées, le temps de traitement des messages de requête et la règle de tri utilisée pour planifier l'activité du contrôleur central.

L'évaluation de configurations plus générales et décorréliées des systèmes de production existants représente également une part importante de notre travail de simulation. Elle a pour but de montrer les effets des différentes règles de tri sur les enchaînements successifs du cycle opératoire de chacune des machines et sur le parcours des pièces produites sur le procédé configuré. Il s'agit de mettre en évidence les situations d'attente des machines de production liées au temps de réponse du contrôleur, à l'absence de pièce à l'entrée ou à l'impossibilité de relâcher une pièce au terme de son traitement en raison d'une saturation de l'espace d'encours. Cette partie de la phase de simulation permet aussi de marquer les différences du niveau de performances entre les règles de tri préemptives et non-préemptives. Elle correspond à un travail prospectif destiné à évaluer des configurations de systèmes imposant des durées importantes de traitement de messages de requête. La part de traitement informationnel étant ainsi accrue par rapport aux systèmes actuels sur lesquels nous nous sommes d'abord basés.

Les résultats de simulation présentés dans ce chapitre correspondent à une synthèse des principales informations obtenues. Ils sont destinés à montrer les aspects majeurs du comportement des systèmes simulés et à mettre en évidence l'impact du contrôleur sur la cadence de production. L'ensemble des enseignements dégagés de la phase de simulation permettent d'exhiber une borne sur la productivité liée à l'utilisation d'un contrôleur en tant que ressource partagée.

5.1 Choix des configurations évaluées et des résultats présentés

Pour l'analyse des systèmes de production existants, la phase de simulation est menée en faisant varier le temps de traitement des messages de requête par le contrôleur central entre 100 ms et 1 000 ms. Ces valeurs sont choisies pour étudier à la fois des situations d'accroissement et de réduction de la vitesse de traitement des messages de requête par le contrôleur par rapport au délai moyen de 500 ms constaté par mesure sur les procédés ayant servis de base à nos travaux.

Les règles de tri *FIFO*, *LIFO*, *RND*, *SPT*, *LPT*, *Push* et *Pull* retenues pour ordonnancer l'activité du contrôleur dans le cadre de la phase de simulation sont d'abord abordées dans un contexte de non-préemption. L'impact des règles *FIFO*, *SPT*, *LPT*, *Push* et *Pull* a ensuite été étudié en intégrant le mécanisme de préemption au modèle. Le bilan d'évaluation qui en découle n'a pas permis de dégager des différences significatives par rapport aux variantes non-préemptives préalablement implémentées. La similitude des premiers résultats obtenus nous amène à ne présenter que les données recueillies dans un contexte non-préemptif. Enfin, nous précisons également que les temps de simulation sont nettement augmentés lorsque le mécanisme de préemption est intégré au modèle.

Les relevés présentés sont obtenus lorsque les systèmes simulés sont en régime stationnaire. Dans notre cas, la date de fin du régime transitoire correspondant à la montée en cadence des procédés est conditionnée par différents critères tels que la durée opératoires des machines, le délai de traitement des messages de requête par le contrôleur et le nombre de machines déclarées sur le système. De manière évidente, le passage en régime stationnaire survient le plus tard pour les configurations incluant le plus grand nombre de machines. Aussi, nous avons fixé une période d'étude en régime stationnaire commune à l'ensemble des systèmes simulés à partir des configurations à trente machines. Différentes méthodes existent pour évaluer le passage d'un régime transitoire à un régime stationnaire pour un système donné. Pour des raisons de simplicité de mise en œuvre, nous avons retenu et utilisé la méthode de Welch [Welch, 1983a] [Welch, 1983b] dont le principe général est présenté en annexe B. Il s'agit d'une technique couramment utilisée et basée sur une analyse graphique pour estimer la date d'établissement du régime stationnaire.

Les simulations prospectives sont basées sur une évaluation à partir des mêmes règles de tri que pour les systèmes existants. Dans ce cas, nous comparons les effets obtenus sur le makespan avec les mécanismes de préemption et de non-préemption pour différentes configurations de systèmes constitués de quatre machines. Seuls, les résultats pertinents obtenus avec les règles de tri *FIFO*, *LPT* et *Pull* sont présentés.

5.2 Définition et évaluation de la borne mécanique

La cadence de production d'un système flow-shop, exprimée par le délai τ d'inter-sorties de pièces, est dans le cas général fixée par le délai de traitement des pièces par la machine la plus lente. Notre étude consiste à évaluer le taux de dégradation induit par la mise en œuvre d'un contrôleur central pour coordonner l'activité des machines. Nous comparons donc les résultats obtenus avec une coordination par un contrôleur avec ceux imposés par la machine la plus lente et que nous considérons a priori comme le meilleur cas pouvant être atteint. Aussi, nous utilisons le terme de *borne mécanique* pour exprimer cette situation optimale.

Le tableau 5.1 exprime le délai d'inter-sorties de pièces correspondant à la borne mécanique pour les différentes configurations simulées dont les résultats sont présentés, soit $\max_{M_i \in \mathcal{M}} \{\theta_i\}$.

Borne mécanique (ms)		Nombre de machines					
		2	5	8	12	20	30
Temps de traitement d'un message de requête (ms)	100	7 490	7 490	7 590	9 390	9 490	9 490
	200	7 690	7 690	7 790	9 590	9 690	9 690
	300	7 890	7 890	7 990	9 790	9 890	9 890
	400	8 090	8 090	8 190	9 990	10 090	10 090
	500	8 290	8 290	8 390	10 190	10 290	10 290
	600	8 490	8 490	8 590	10 390	10 490	10 490
	700	8 690	8 690	8 790	10 590	10 690	10 690
	800	8 890	8 890	8 990	10 790	10 890	10 890
	900	9 090	9 090	9 190	10 990	11 090	11 090
	1 000	9 290	9 290	9 390	11 190	11 290	11 290

TAB. 5.1: Évaluation de la borne mécanique

Les délais exprimés sont calculés à partir des données indiquées par le tableau 4.6 présenté au chapitre précédent et définissant la somme des temps de pré-assemblage, d'assemblage et de post-assemblage. Ils intègrent les temps de traitement des messages de requête par le contrôleur central et représentent ainsi les temps opératoires effectifs.

5.3 Analyse de la cadence de production

L'analyse de la cadence de production, donnée par l'équation 3.3, est un moyen simple et direct pour évaluer le degré de productivité d'un système manufacturier. Pour cette raison, le makespan constitue souvent le critère permettant d'évaluer le niveau de performances d'un procédé industriel. Aussi, les résultats de synthèse que nous présentons dans cette section concernent le délai τ d'inter-sorties de pièces que nous comparons à la borne mécanique. En effet, une perte de productivité est a priori constatée dès lors que le système ne produit pas à la vitesse de la machine la plus lente.

5.3.1 Comparaison des règles de tri

Les résultats obtenus concernant la cadence de production sont présentés par le délai moyen d'inter-sorties de pièces en fonction de la règle de tri utilisée et des temps de traitement des messages de requête par le contrôleur pour chacune des configurations de systèmes étudiées.

Les figures 5.6 à 5.1 présentent sous la forme d'histogrammes une synthèse des résultats relevés sur la cadence de production. L'écart à la borne mécanique croît de façon évidente lorsque le nombre de machines déclarées et le temps de traitement des messages de requête augmentent. Toutefois, aucune perte de productivité significative n'est constatée pour les configurations à deux et cinq machines de même que pour tous les systèmes lorsque le temps de traitement des messages de requête égal à 100 ms. Les résultats de simulation obtenus pour la configuration à deux machines montrent même que le makespan est optimal dans tous les cas. Le seuil d'observation des premiers signes de dégradation des performances varie selon les configurations. Il peut être graphiquement et de façon approximative situé à 200 ms pour un système à trente machines de production, à 300 ms pour vingt machines, à 400 ms pour douze machines et à 500 ms pour huit machines.

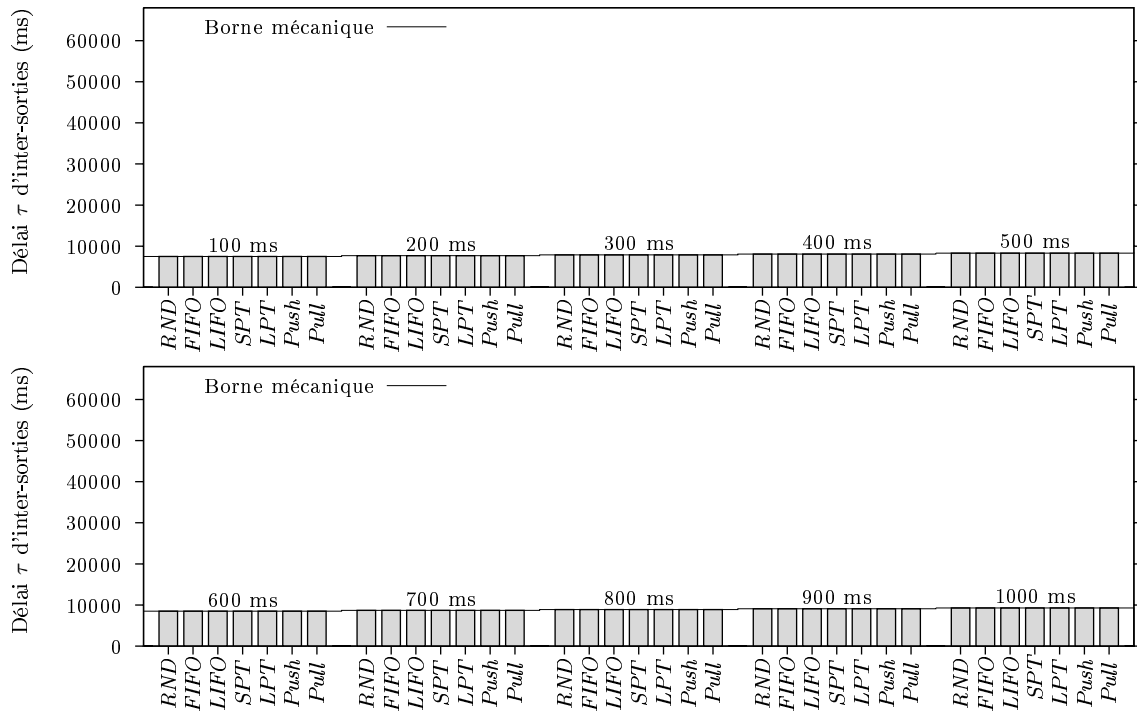


FIG. 5.1: Cadence de production pour un système à 2 machines

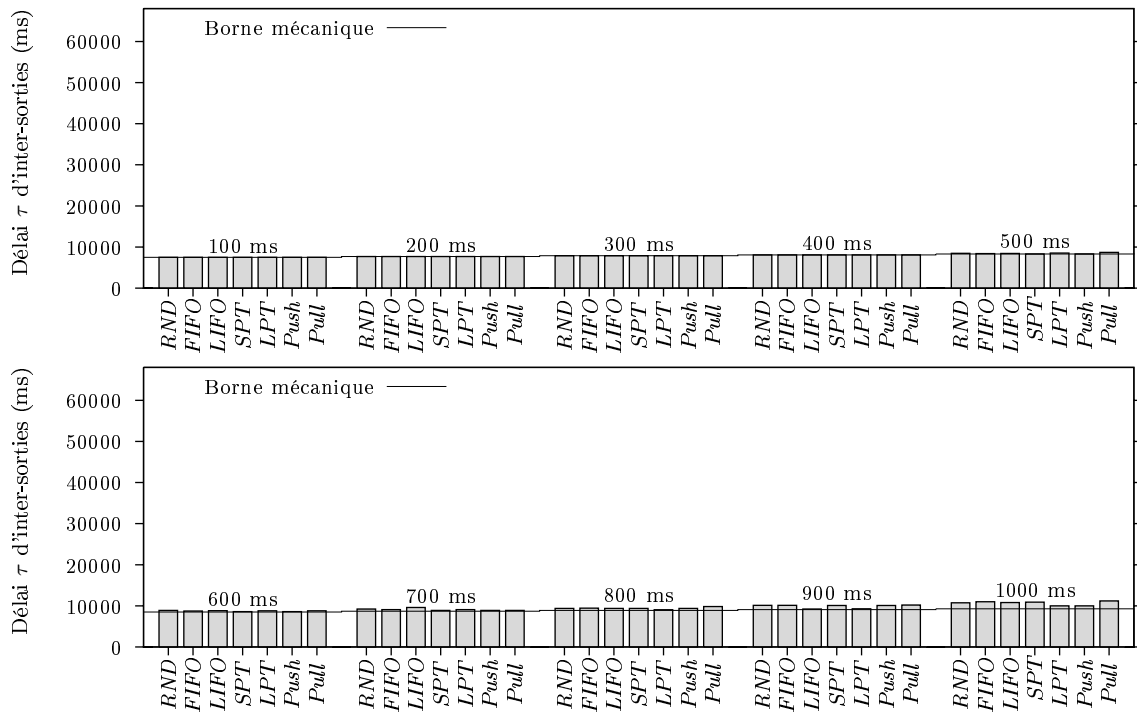


FIG. 5.2: Cadence de production pour un système à 5 machines

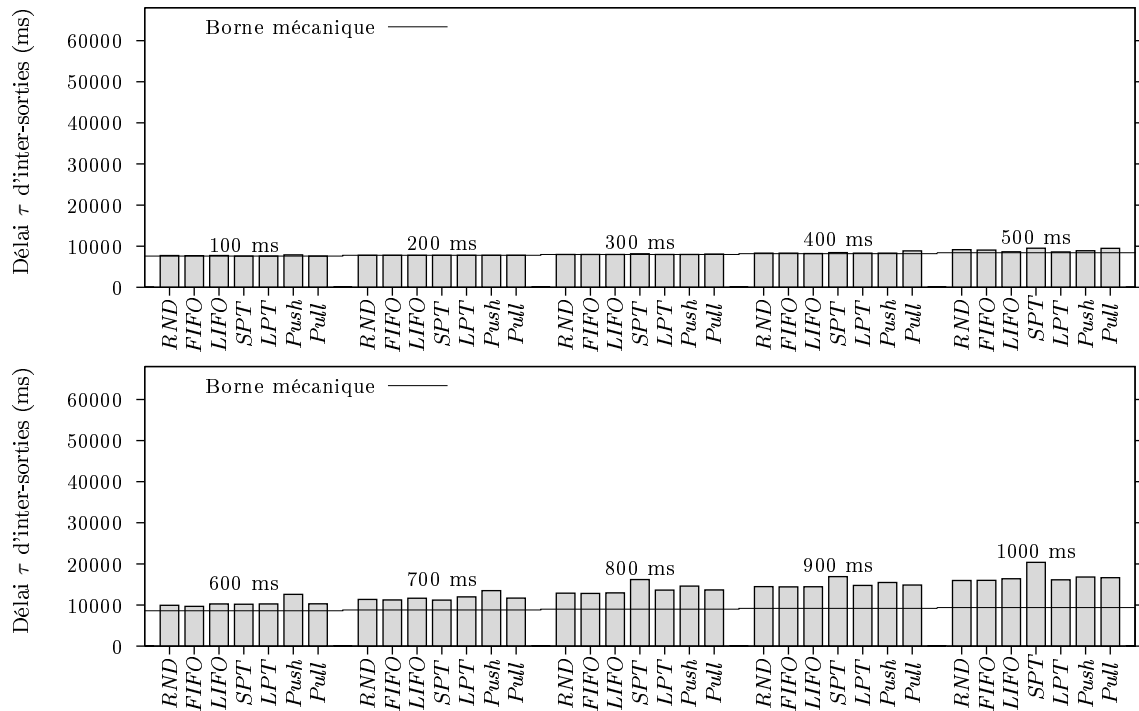


FIG. 5.3: Cadence de production pour un système à 8 machines

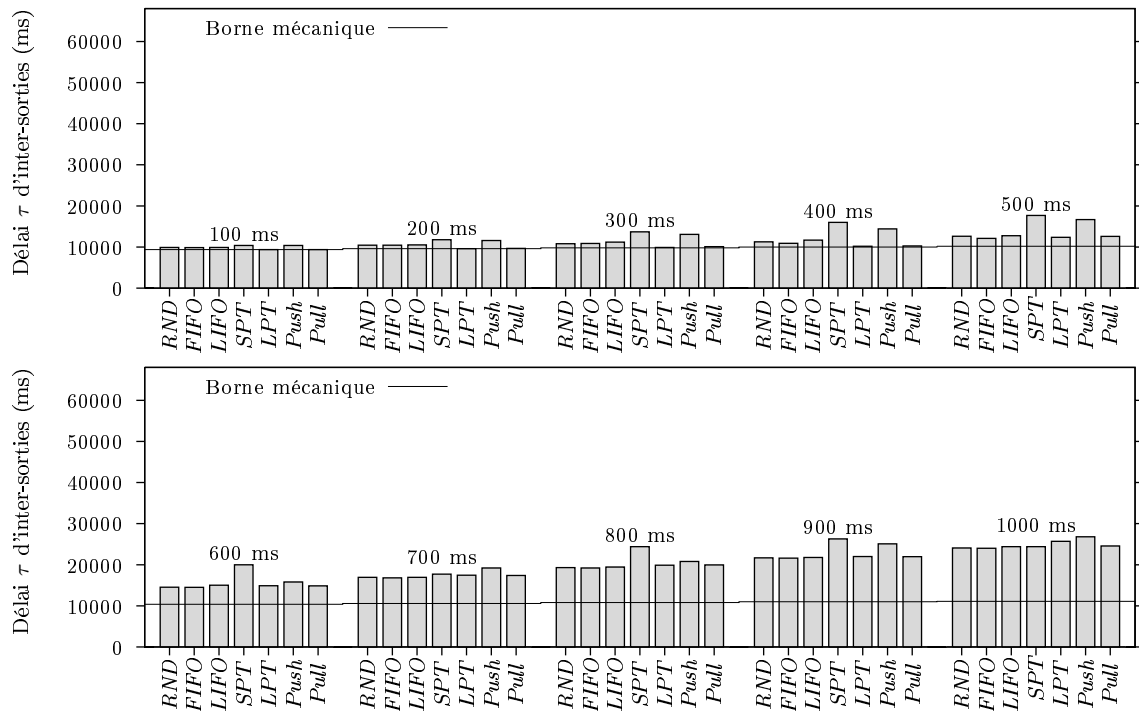


FIG. 5.4: Cadence de production pour un système à 12 machines

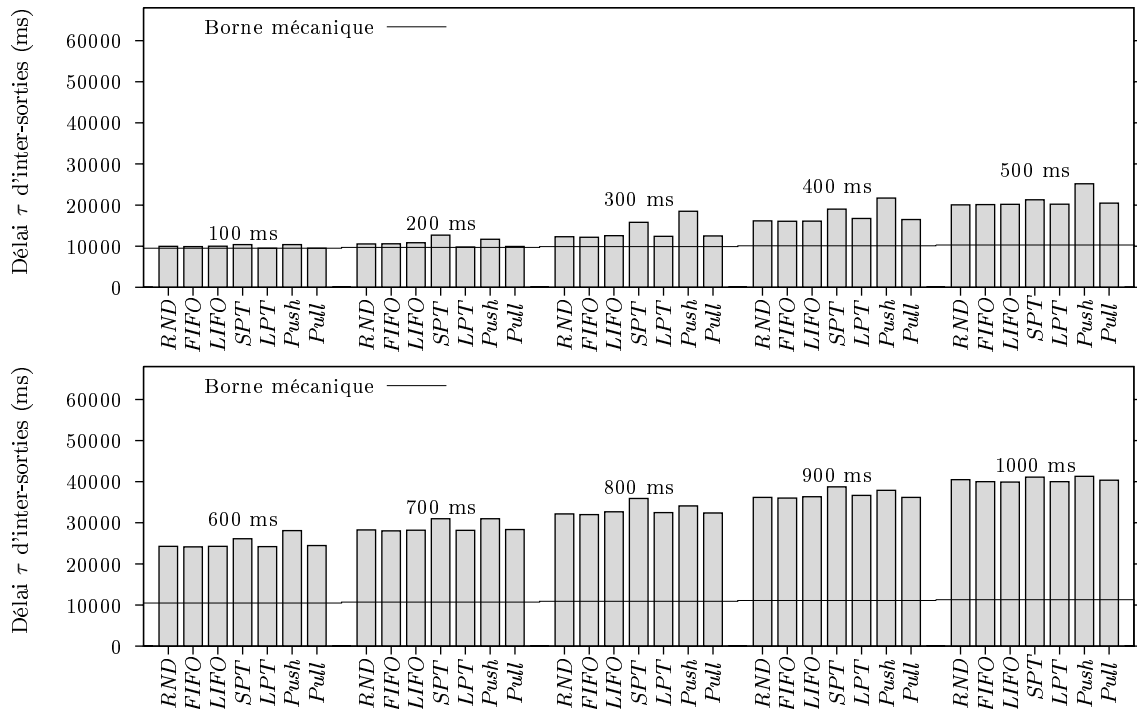


FIG. 5.5: Cadence de production pour un système à 20 machines

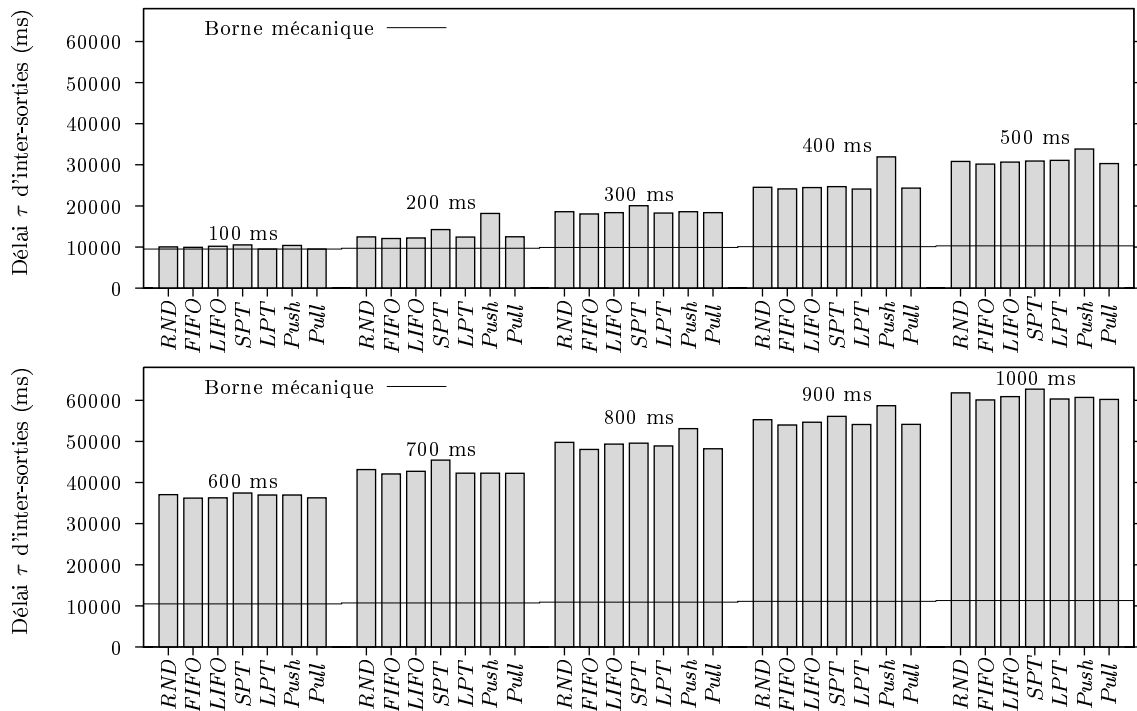


FIG. 5.6: Cadence de production pour un système à 30 machines

De manière générale, nous constatons que les règles de tri *SPT* et *Push* donnent dans la plupart des cas les plus mauvais résultats. Intuitivement, nous pouvons avancer l'hypothèse qu'elles ont en effet pour conséquence de forcer l'accumulation de pièces sur le système de production et donc de favoriser les situations d'encours. Quelle que soit la règle de tri utilisée parmi *RND*, *FIFO* et *LIFO*, les résultats produits sont sensiblement identiques.

5.3.2 Mise en évidence de l'impact du contrôleur

L'augmentation du nombre de machines et du temps de traitement des messages de requête influe directement sur la charge du contrôleur central dont l'accroissement s'accompagne d'une dégradation de la productivité des systèmes. Le tableau 5.2 montre l'affaiblissement des performances en présentant les écarts de productivité relevés par rapport à la borne mécanique. Le degré de productivité comparé à la borne mécanique indiqué pour chaque configuration est celui obtenu par la règle de tri ayant donné le résultat le plus favorable.

Degré de productivité (%)		Nombre de machines					
		2	5	8	12	20	30
Temps de traitement d'un message de requête (ms)	100	100.00	100.00	100.00	100.00	100.00	100.00
	200	100.00	100.00	100.00	100.00	98.88	80.34
	300	100.00	100.00	100.00	98.89	81.36	54.79
	400	100.00	100.00	99.88	98.21	63.53	41.87
	500	100.00	100.00	97.56	84.23	51.33	34.69
	600	100.00	98.84	88.94	71.69	43.43	28.98
	700	100.00	97.75	78.48	63.04	38.13	25.62
	800	100.00	98.34	70.23	56.20	34.03	22.66
	900	100.00	98.80	63.82	50.88	30.81	20.67
	1 000	100.00	92.90	58.79	46.63	28.29	18.85

TAB. 5.2: Degré de productivité atteint par rapport à la borne mécanique

Nous remarquons que le taux de dégradation de la productivité des systèmes progresse en fonction de la charge soumise au contrôleur et indépendamment de la borne mécanique. À titre d'exemple, malgré que les configurations à vingt et trente machines présentent la même borne mécanique imposée par la dix-septième machine, le taux de

dégradation de la productivité est plus important dans le cas de la configuration à trente machines. Dans le cas de ces configurations, le makespan n'est donc pas fixé par la machine la plus lente.

Pour la plupart des configurations de systèmes simulés, l'impact du contrôleur sur les performances est alors mis en évidence. Il se traduit par un allongement du délai d'inter-sorties de pièces lié à la concurrence d'accès des machines au contrôleur et aux situations de suspension de leur cycle opératoire durant l'attente d'un message de réponse.

5.4 Observation du temps de réponse du contrôleur

Le temps de réponse du contrôleur central aux requêtes émises par les machines de production est un élément important qu'il est nécessaire d'observer. Les valeurs moyennes relevées au cours de la phase de simulation sont exprimées par les histogrammes de la figure 5.7. Les graphiques intègrent les données collectées pour l'ensemble des configurations simulées.

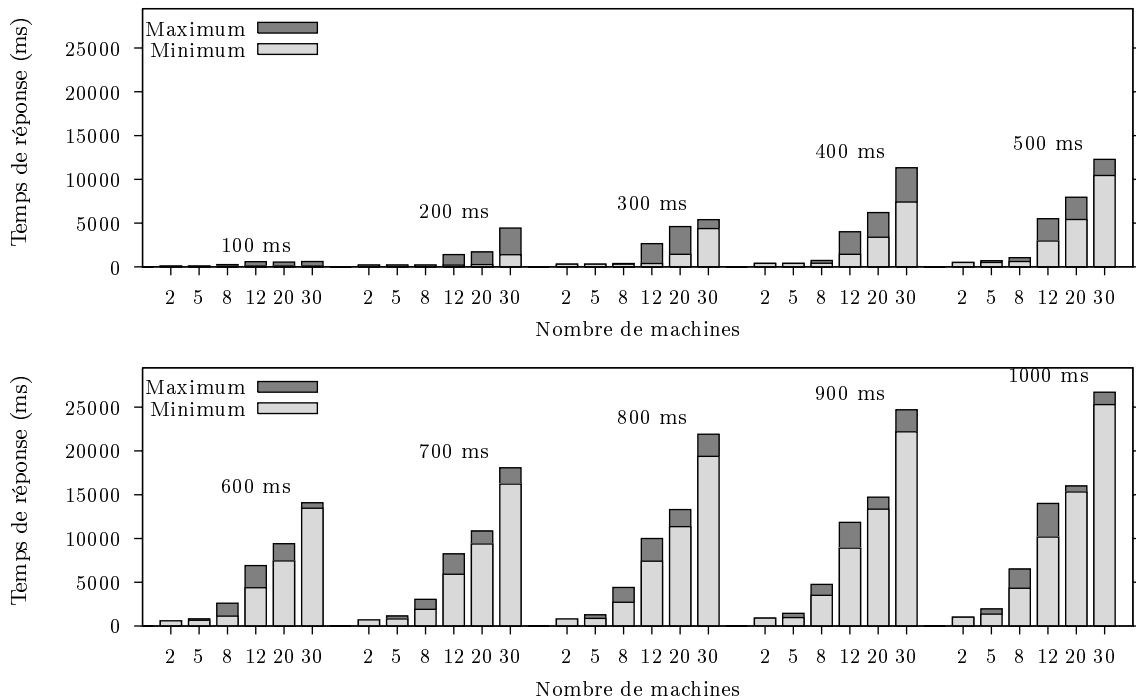


FIG. 5.7: Visualisation des temps de réponse du contrôleur

Les résultats montrent, conjointement pour chacun des temps de traitement de messages de requête paramétrés pour le contrôleur, les résultats obtenus avec la plus favorable et la plus défavorable des règles de tri. Le temps de réponse du contrôleur croît de manière évidente lorsque le nombre de machine et le temps de traitement des messages de requête augmentent. Dans le cas de la configuration la plus lourde, soit trente machines déclarées et une durée de 1000 ms nécessaire au traitement d'un message de requête, le temps de réponse moyen du contrôleur est de l'ordre de 25 s. Ce constat dénote d'un grand nombre de messages de requête en attente à chaque instant dans le tampon de communication et donc d'une activité permanente du contrôleur. Par contre, lorsque seulement deux machines sont déclarées sur le système, les indications relevées montrent que, quelle que soit la règle de tri utilisée, le temps de réponse est équivalent à la durée nécessaire au traitement d'un message de requête. Les messages ne font alors l'objet d'aucune attente dans le tampon de communication avant leur prise en charge par le contrôleur et le système produit à une cadence qui s'établit à la borne mécanique.

L'analyse des résultats de simulation laisse également apparaître que, dans le cas extrême correspondant à la configuration à trente machines avec une durée de 100 ms nécessaire au traitement d'un message de requête, le temps de réponse moyen du contrôleur varie entre 100 ms et plus de six fois cette valeur. Le choix de la règle de tri utilisée peut donc influencer de manière importante sur le temps de réponse du contrôleur.

5.5 Cadence imposée par le contrôleur

Les résultats de simulation démontrent l'influence du contrôleur central sur les performances des systèmes. Pour la plupart des configurations étudiées, ils permettent de constater que la cadence des procédés n'atteint pas la borne mécanique. Le contrôleur central représente alors, dans ces cas précis, la ressource critique du système simulé. Il constitue ainsi le goulet d'étranglement.

5.5.1 Formulation d'une borne de contrôle

Le contrôleur est une ressource particulière du système. Sa charge d'activité est conditionnée par le cumul des opérations distantes affectées à la gamme opératoire de chacune des machines. Dans une certaine mesure, nous pouvons l'appréhender comme

une ressource classique au même titre que les machines de production dont le temps de rétention des pièces serait établi en fonction de sa charge, soit :

$$\sum_{M_i \in \mathcal{M}} \theta_i^* \quad (5.1)$$

Il est alors possible d'abstraire le contrôleur en le considérant comme toutes les autres machines implantées successivement sur le système. La figure 5.8 présente un raffinement du modèle d'un système flow-shop contrôlé présenté au chapitre 3 par la figure 3.8.

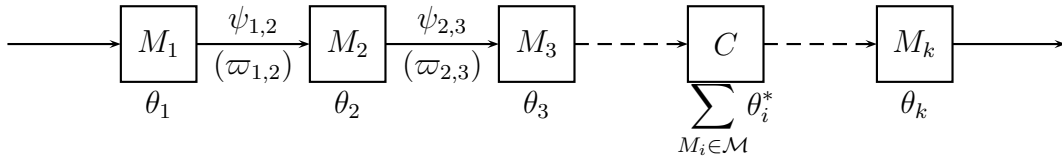


FIG. 5.8: *Simplification du modèle des systèmes contrôlés*

Il est admis que chacune des machines présente une borne sur la cadence de production limitant la valeur du makespan pour le système flow-shop étudié et que la borne mécanique est établie par la plus lente. Nous pouvons alors évaluer le délai minimum d'inter-sorties de pièces fixé par le contrôleur central. Le tableau 5.3 présente les différentes valeurs calculées de la borne de contrôle.

Borne de contrôle (ms)		Nombre de machines					
		2	5	8	12	20	30
Temps de traitement d'un message de requête (ms)	100	400	1 000	1 600	2 400	4 000	6 000
	200	800	2 000	3 200	4 800	8 000	12 000
	300	1 200	3 000	4 800	7 200	12 000	18 000
	400	1 600	4 000	6 400	9 600	16 000	24 000
	500	2 000	5 000	8 000	12 000	20 000	30 000
	600	2 400	6 000	9 600	14 400	24 000	36 000
	700	2 800	7 000	11 200	16 800	28 000	42 000
	800	3 200	8 000	12 800	19 200	32 000	48 000
	900	3 600	9 000	14 400	21 600	36 000	54 000
	1 000	4 000	10 000	16 000	24 000	40 000	60 000

TAB. 5.3: *Évaluation de la borne de contrôle*

Lorsque la cadence de production d'un système est fixée par la borne mécanique,

le délai d'inter-sorties de pièces s'établit exactement à la valeur du délai de rétention de pièces par la machine la plus lente. La borne de contrôle impose une limite sur la cadence de production dès lors qu'elle devient supérieure à la borne mécanique du système considéré :

$$\sum_{M_i \in \mathcal{M}} \theta_i^* > \max_{M_i \in \mathcal{M}} \{\theta_i\} \quad (5.2)$$

Dans le cas où la borne de contrôle prévaut sur la borne mécanique et que le contrôleur central devient ainsi le goulet d'étranglement du système, le délai d'inter-sorties de pièces n'atteint pas systématiquement la limite imposée par la borne. L'écart obtenu varie selon la règle de tri utilisée pour organiser l'activité du contrôleur. La borne de contrôle représente alors une situation optimale qui ne peut être atteinte que dans certains cas en fonction de la configuration du système et de la règle de tri utilisée.

En effet, contrairement au contexte d'exécution du cycle opératoire des machines qui ne retiennent qu'une seule pièce à la fois, le contrôleur est soumis à une concurrence d'accès qui impose une sélection de la machine à servir. Les messages de requête peuvent donc faire l'objet d'une attente dans le tampon communication préalablement à leur prise en charge et leur traitement par le contrôleur ayant pour conséquence un allongement de la durée de rétention des pièces concernées sur les machines. L'atteinte de la borne de contrôle n'est donc pas garantie dans le cas où le contrôleur est la ressource critique.

5.5.2 Comparaison des bornes sur la cadence de production

Les histogrammes des figures 5.6 à 5.1 montrent que le délai d'inter-sorties de pièces s'éloigne de la borne mécanique lorsque le nombre de machines et le temps de traitement des messages de requête par le contrôleur central augmentent. Nous savons que l'écart avec la borne mécanique existe dès lors que le contrôleur central constitue la ressource critique du système flow-shop et qu'il est directement lié à la somme des temps de traitement des messages de requête nécessaires à la production d'une pièce. Il est alors intéressant de comparer la cadence de production atteinte par les systèmes simulés conjointement à la borne mécanique et à la borne de contrôle afin mettre en évidence l'impact du contrôleur sur le makespan et de vérifier graphiquement les effets de la borne de contrôle. Nous reprenons alors sur les figures 5.14 à 5.9 les résultats sur la cadence de production comparés aux deux bornes.

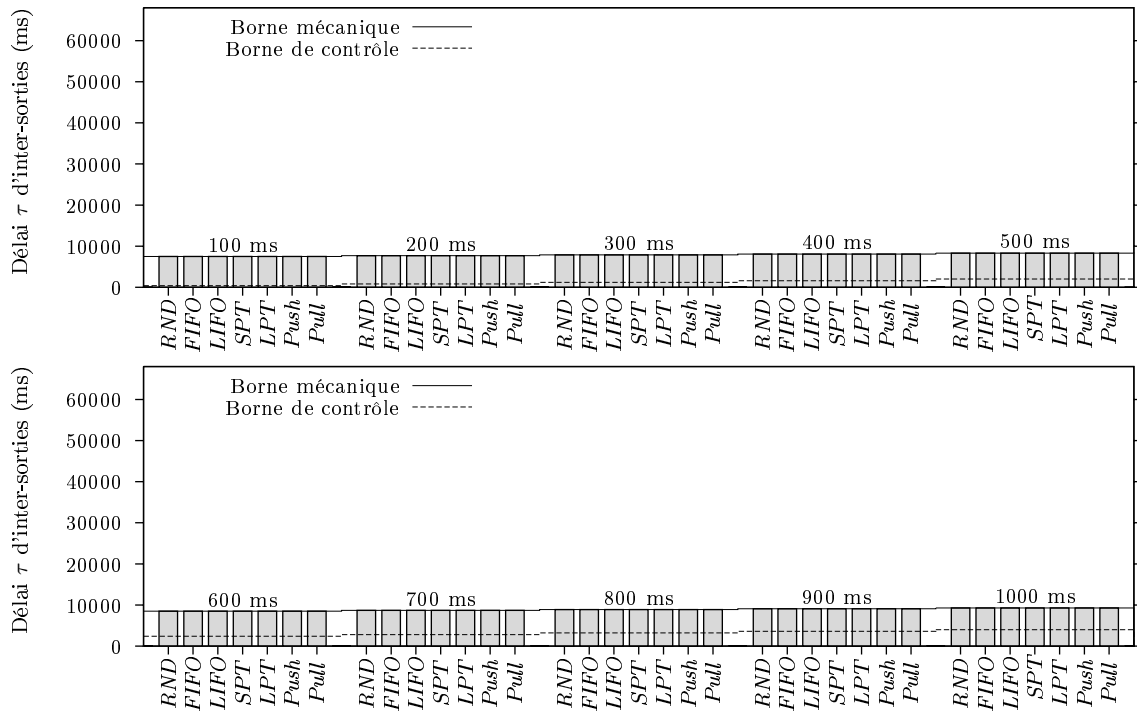


FIG. 5.9: Comparaison des bornes pour un système à 2 machines

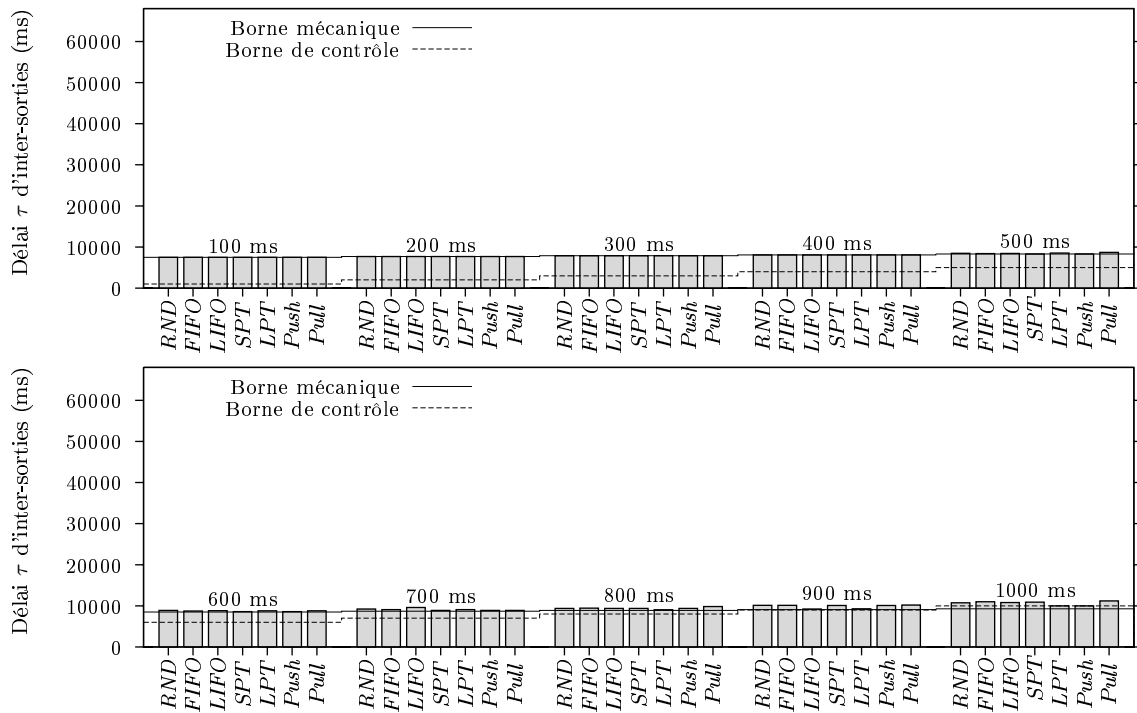


FIG. 5.10: Comparaison des bornes pour un système à 5 machines

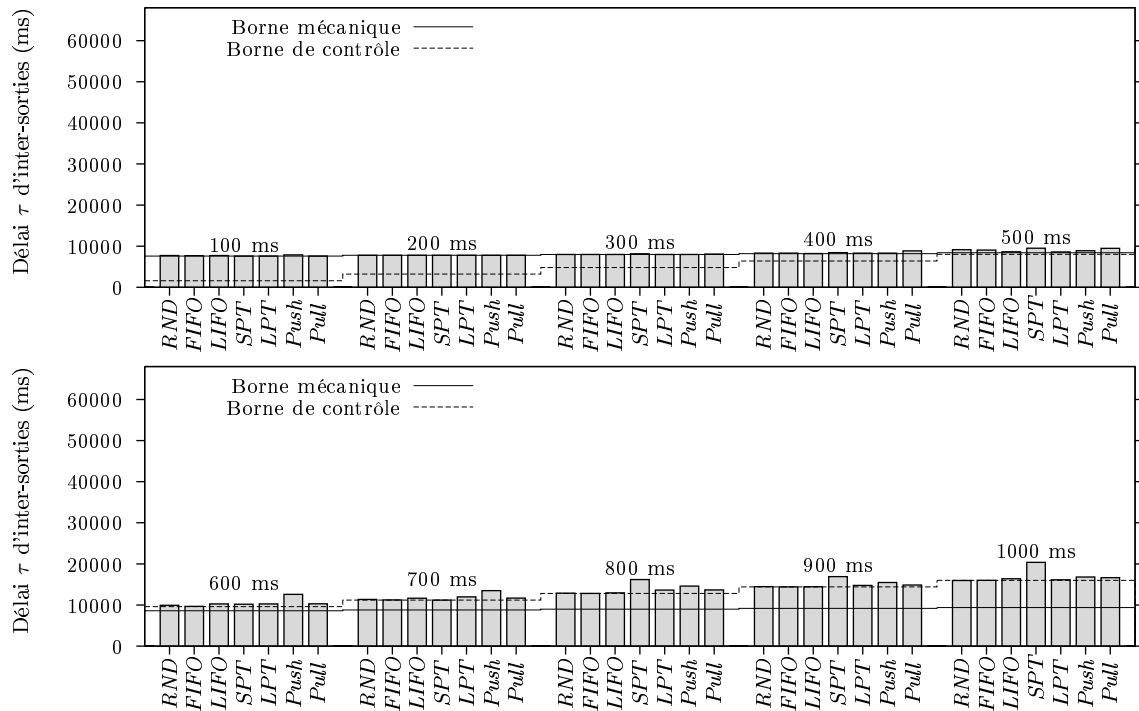


FIG. 5.11: Comparaison des bornes pour un système à 8 machines

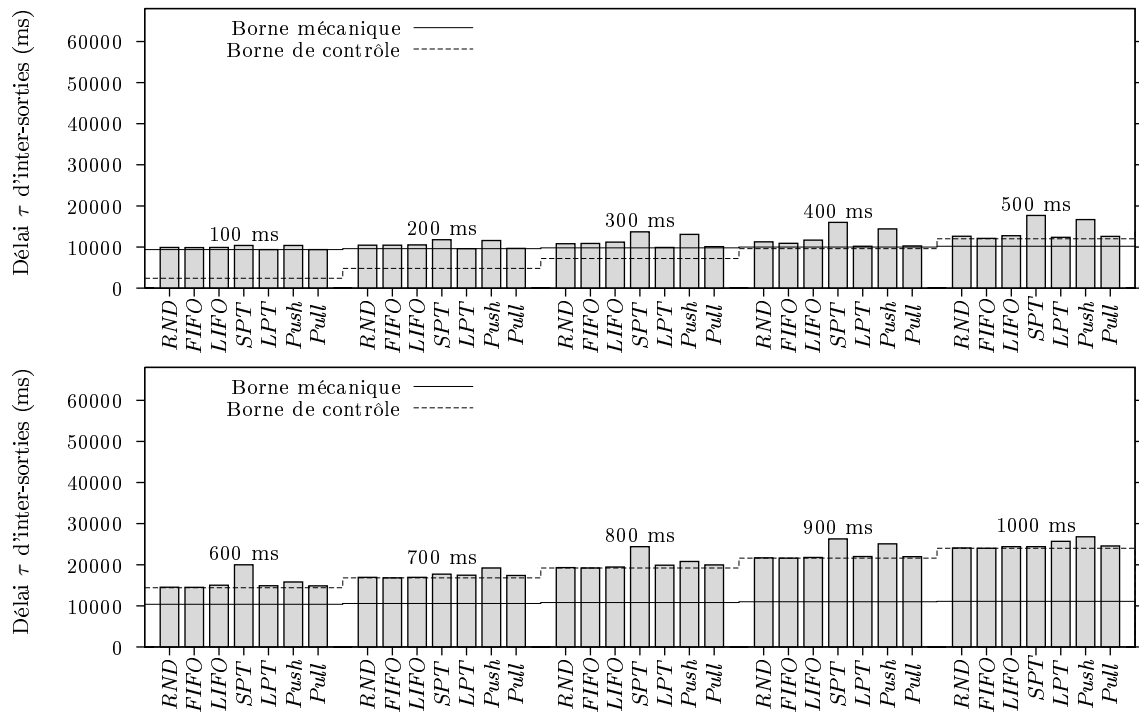


FIG. 5.12: Comparaison des bornes pour un système à 12 machines

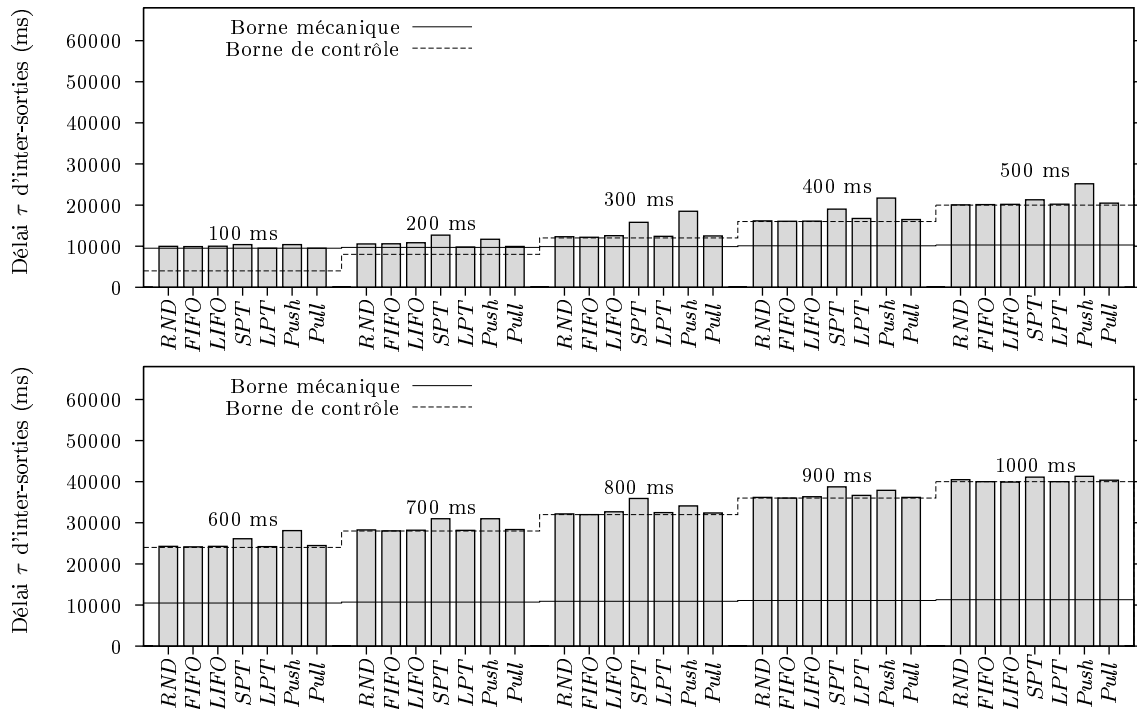


FIG. 5.13: Comparaison des bornes pour un système à 20 machines

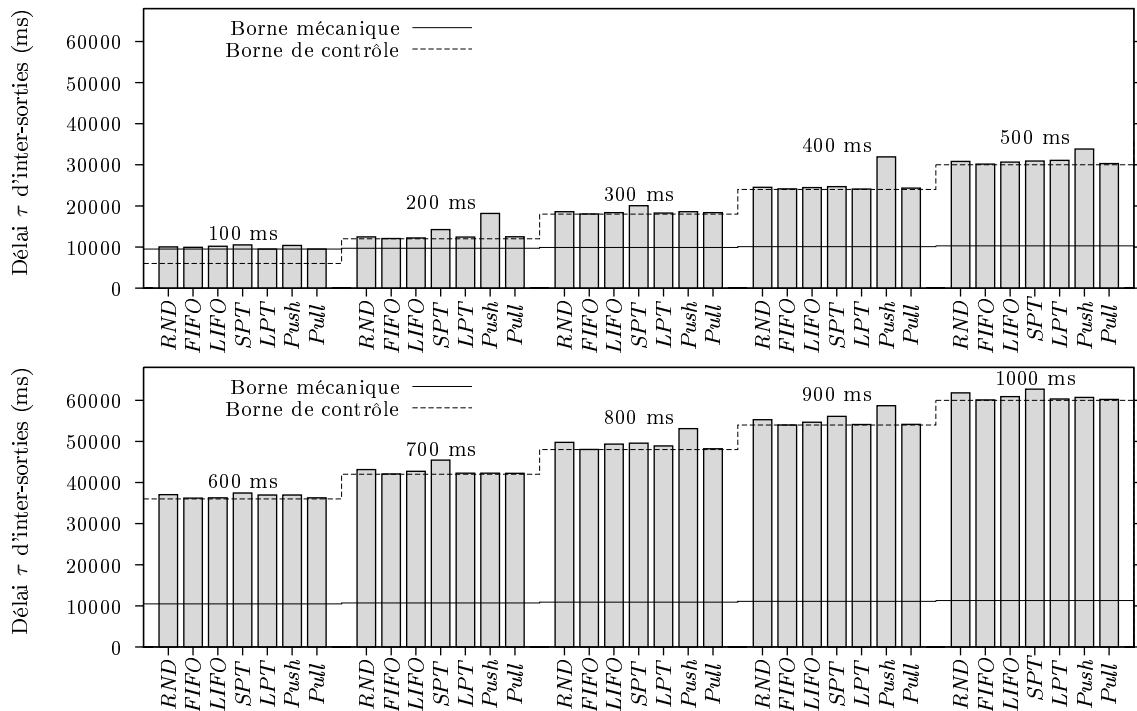


FIG. 5.14: Comparaison des bornes pour un système à 30 machines

Les graphiques confirment l'influence du contrôleur central sur le makespan. Ils montrent que le délai d'inter-sorties de pièces est souvent conditionné par la charge du contrôleur et que la borne de contrôle n'est pas systématiquement atteinte lorsqu'elle prévaut sur la borne mécanique.

Nous montrons sur la figure 5.15, à travers deux graphiques, l'évolution comparative des bornes mécanique et de contrôle en faisant varier le nombre de machines puis le délai de traitement des messages de requête par le contrôleur. Nous nous basons d'abord sur le cas de chacune des configurations évaluées pour lesquelles le temps de traitement des messages de requête est paramétré à 500 ms. Nous présentons ensuite les résultats obtenus lors de la simulation des systèmes constitués de douze machines.

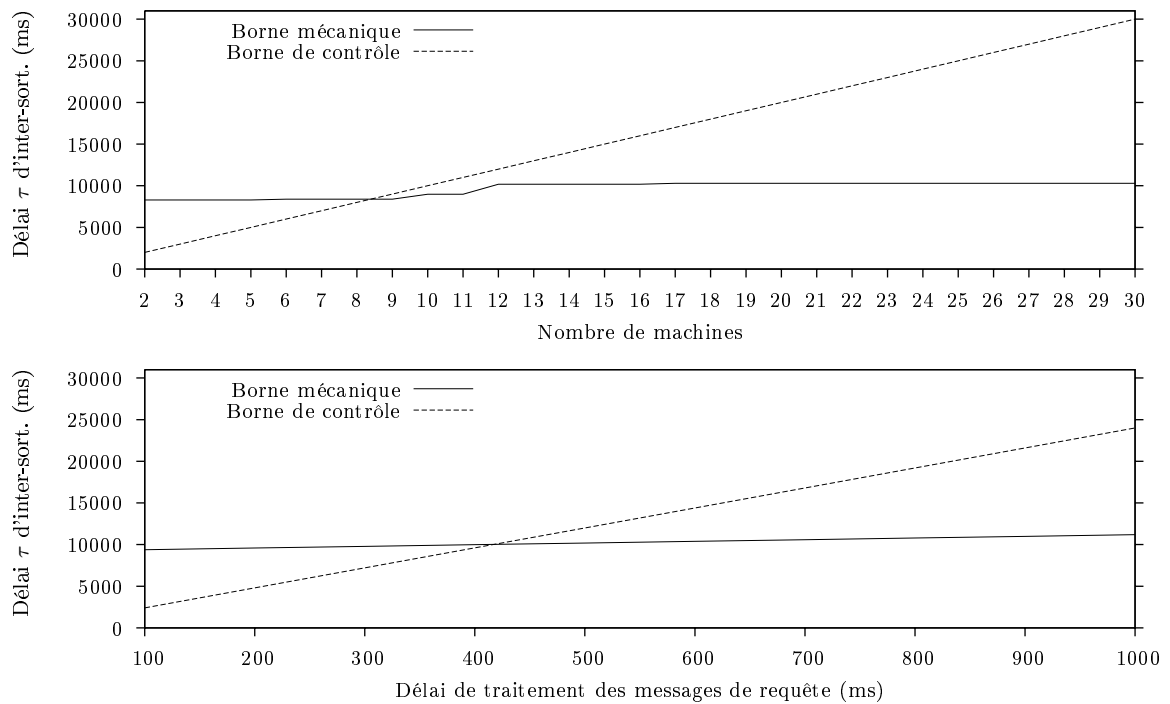


FIG. 5.15: *Évolution des bornes mécanique et de contrôle*

Le premier exemple montre, qu'avec un temps de traitement des messages de requête paramétré à 500 ms, la borne mécanique prévaut tant que le système n'est pas constitué de plus de huit machines. Le second exemple permet de constater, que pour un système à douze machines, la borne de contrôle prévaut lorsque le temps de traitement des messages de requête se situe au-delà d'une valeur sensiblement supérieure à 400 ms.

5.5.3 Redéfinition de la condition d'optimalité et de la notion de latence

Le délai moyen τ d'inter-sorties de pièces définit la cadence atteinte par le procédé. Pour un système flow-shop non-contrôlé, la borne mécanique correspond à une valeur optimale pouvant être aisément estimée et qui, dans le cas courant, ne dépend que du temps de rétention des pièces sur la machine de production la plus lente du système. L'intégration d'un contrôleur central à un système flow-shop ne permet plus de garantir cette condition d'optimalité. En effet, les résultats obtenus par la phase de simulation mettent en évidence le fait que le contrôleur central puisse constituer le goulet d'étranglement d'un système flow-shop contrôlé. Lorsque la borne de contrôle prévaut sur la borne mécanique, le procédé ne peut en aucun cas produire à la vitesse de la machine la plus lente. Nous devons alors limiter l'évaluation du délai moyen d'inter-sorties de pièces à une valeur minimale correspondant à une borne sur la cadence de production que nous notons τ_{max} :

$$\tau_{max} = \max \left\{ \max_{M_i \in \mathcal{M}} \{\theta_i\}, \sum_{M_i \in \mathcal{M}} \theta_i^* \right\} \quad (5.3)$$

Cependant, afin de généraliser l'évaluation de la borne sur la cadence de production, nous devons y intégrer le cas particulier où la ressource critique est constituée par une phase de transfert des pièces entre deux machines. Nous pouvons alors préciser le calcul de la cadence de production donné par l'équation 3.7 en y intégrant la charge d'activité assurée par le contrôleur central :

$$\tau_{max} = \max \left\{ \max_{M_i \in \mathcal{M}} \{\theta_i\}, \max_{M_i \in [M_2, M_k]} \left\{ \frac{\psi_{i-1,i}}{\varpi_{i-1,i}} \right\}, \sum_{M_i \in \mathcal{M}} \theta_i^* \right\} \quad (5.4)$$

À partir de cette équation, il est possible de prédire le niveau optimal de productivité qu'un système flow-shop contrôlé est en mesure d'atteindre. Elle s'applique quel que soit l'élément constituant la ressource critique du procédé, qu'il s'agisse d'une machine de production, du contrôleur central ou d'une phase de transfert des pièces entre deux machines implantées successivement. Lorsque le goulet d'étranglement est constitué par le contrôleur central, l'écart entre le délai moyen τ d'inter-sorties de pièces et

la borne τ_{max} sur la cadence de production dépend directement de la pertinence de la règle de tri utilisée pour gérer l'activité du contrôleur.

De même, la notion de latence d'activité $\bar{\theta}_i$ des machines nécessite d'être redéfinie en tenant compte des conditions optimales d'exécution des systèmes contrôlés en précisant l'équation 3.16 :

$$\forall M_i \in \mathcal{M} : \bar{\theta}_i = \tau_{max} - \theta_i \quad (5.5)$$

La latence d'activité $\bar{\theta}_i$, donnée par l'équation 5.5, définit un délai minimum d'inactivité dont dispose chacune des machines de production entre le début de traitement de deux pièces successives. Elle peut être partiellement ou totalement consommée pour absorber des situations de suspension d'activité des machines dues à un délai avant le début de traitement d'un message de requête par le contrôleur ou à une impossibilité de relâcher une pièce en raison de la saturation de l'espace d'encours.

5.6 Extension des simulations

Les résultats présentés dans les sections précédentes relèvent de la simulation de systèmes de production dont la configuration est basée sur des procédés existants. Des simulations complémentaires ont été menées afin de confirmer les informations dégagées mais aussi de préciser certains aspects du comportement des systèmes flow-shop contrôlés.

Il s'agit en outre de montrer d'une manière plus large, à travers quelques exemples basés sur une gamme opératoire simple, les écarts de performances obtenus par l'utilisation de différentes règles de tri lorsque le cumul des temps opératoires affectés à des opérations distantes est important. Les procédés existants, ayant servi de base à notre étude, ne présentaient en effet qu'une part limitée de traitement distant et informationnel confiée au contrôleur central. Notre objectif est ici d'évaluer le degré de criticité du choix de la règle de tri pour des cas plus généraux. Nous présentons les résultats d'exécution sous la forme d'une série de diagrammes de Gantt.

5.6.1 Choix d'une gamme opératoire et des règles de tri évaluées

Les règles de tri retenues pour ces simulations complémentaires sont celles qui produisent les meilleurs résultats dans le cas des configurations initialement évaluées. Il

s'agit d'abord de *FIFO*, non-préemptive par nature, puis de *LIFO*, *LPT* et *Pull* dou-blement abordées dans les contextes de préemption et de non-préemption du traitement des messages de requête.

La gamme opératoire retenue pour paramétrer le modèle dans le cadre des simula-tions complémentaires est présentée dans le tableau 5.4. Elle se décline en une séquence de douze opérations dont l'affectation est équitablement répartie sur un ensemble de quatre machines. La borne τ_{max} sur la cadence de production est fixée à dix unités de temps par le contrôleur central.

Opérations		M_i	θ_i	$\bar{\theta}_i$	τ_{max}
$\theta_{1,1}$	2	1	7	3	10
$\theta_{1,2}^*$	3				
$\theta_{1,3}$	2				
$\theta_{2,1}$	2	2	8	2	
$\theta_{2,2}^*$	4				
$\theta_{2,3}$	2				
$\theta_{3,1}$	2	3	5	5	
$\theta_{3,2}^*$	1				
$\theta_{3,3}$	2				
$\theta_{4,1}$	2	4	6	4	
$\theta_{4,2}^*$	2				
$\theta_{4,3}$	2				

TAB. 5.4: Gamme opératoire pour les simulations complémentaires

Les autres éléments du paramétrage du modèle concernent le délai de transfert $\psi_{i-1,i}$ des pièces d'une machine vers la suivante et la capacité d'encours $\varpi_{i-1,i}$ disponible entre deux machines successives. Ils sont fixés de manière homogène pour l'ensemble du système simulé à des valeurs respectives de 3 et 1 unités de temps.

5.6.2 Résultats d'exécution

Les figures 5.16 à 5.22 montrent les résultats d'exécution obtenus au cours d'une période de simulation étalée sur 195 unités de temps. Elles permettent de comparer graphiquement le niveau de productivité atteint avec chacune des six règles de tri évaluées.

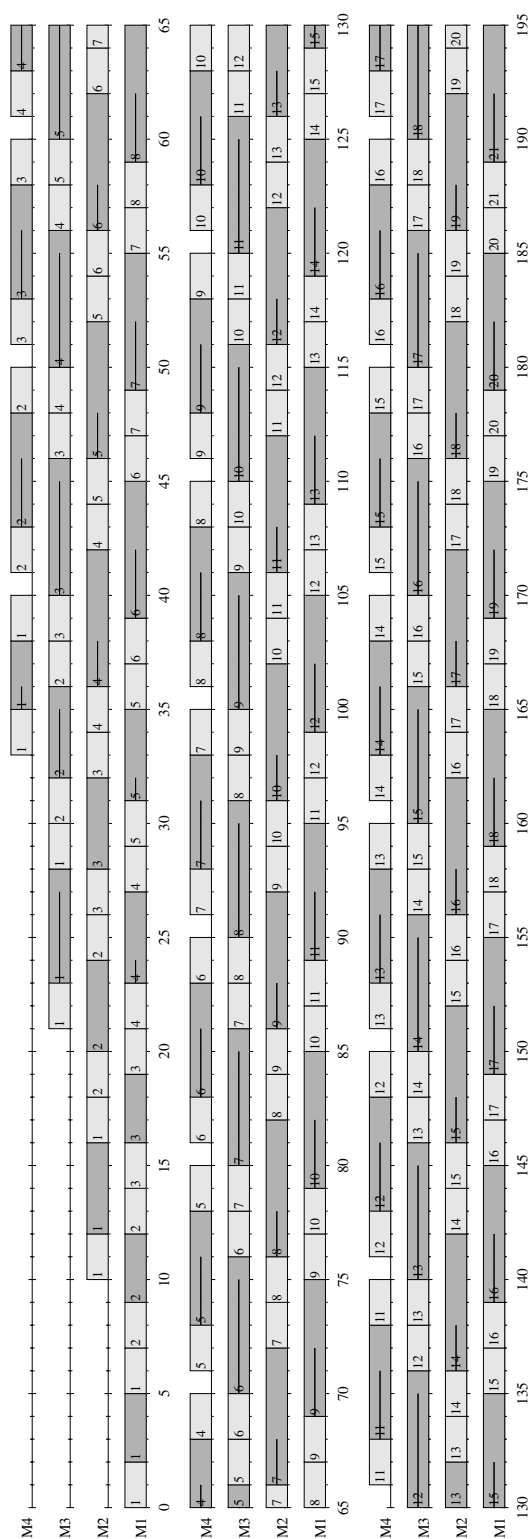


FIG. 5.16: Diagramme d'exécution avec la règle de tri FIFO

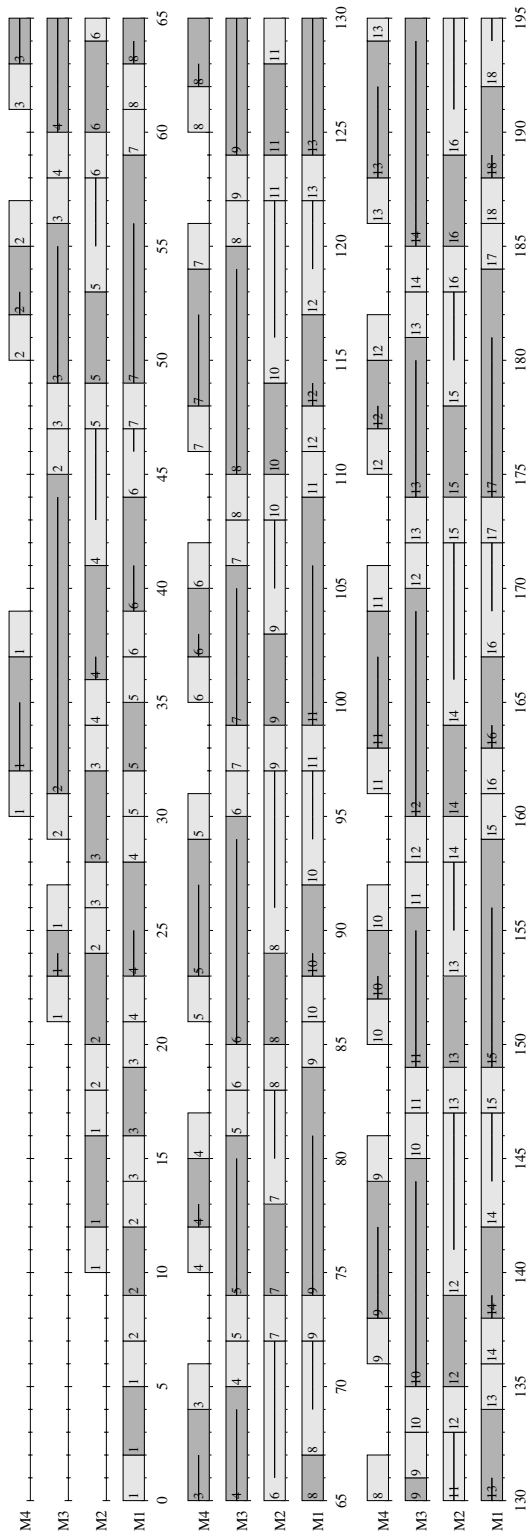


FIG. 5.17: Diagramme d'exécution avec la règle de tri LIFO non-préemptive

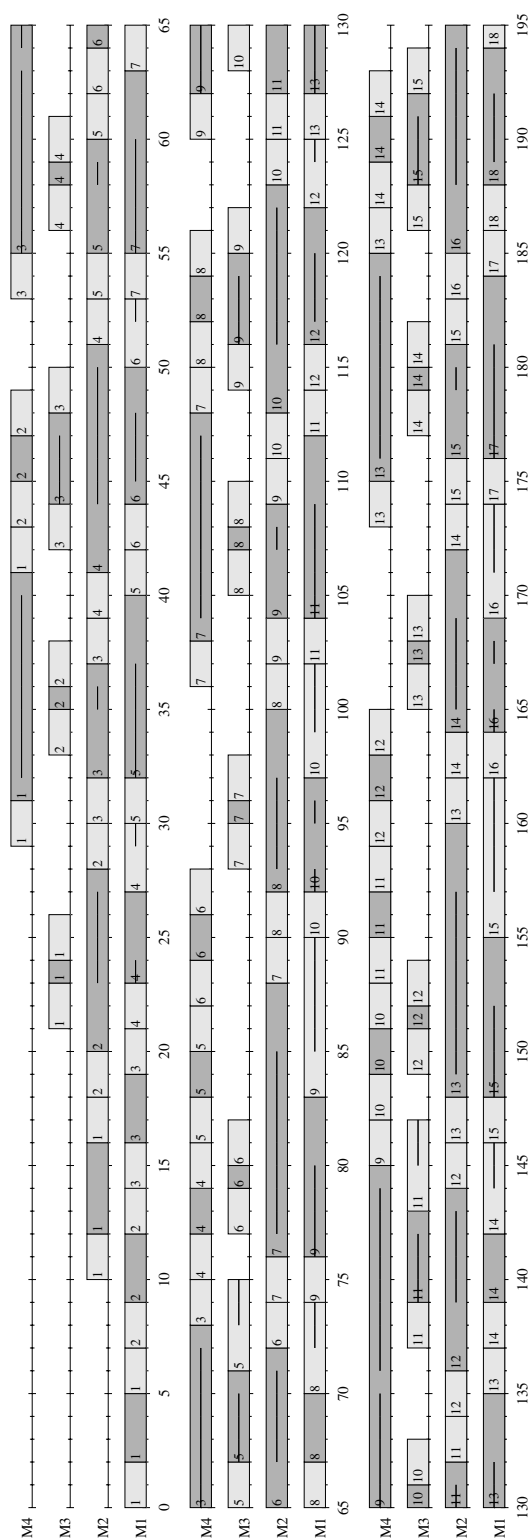


FIG. 5.18: Diagramme d'exécution avec la règle de tri LIFO préemptive

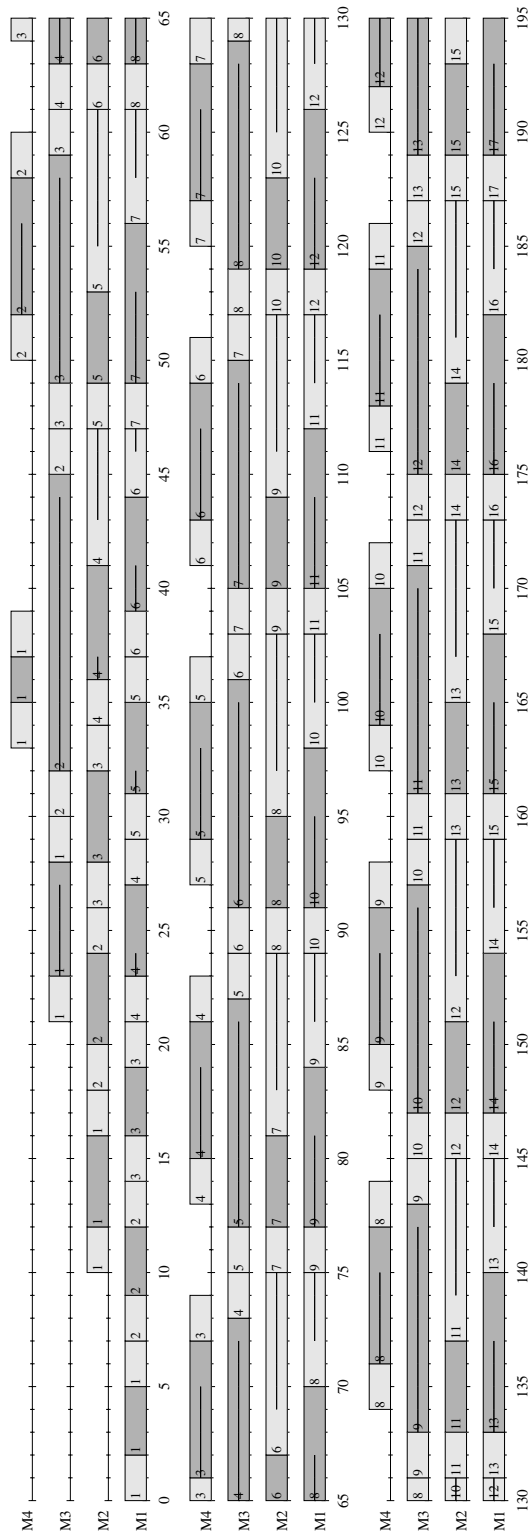


FIG. 5.19: Diagramme d'exécution avec la règle de tri LPT non-préemptive

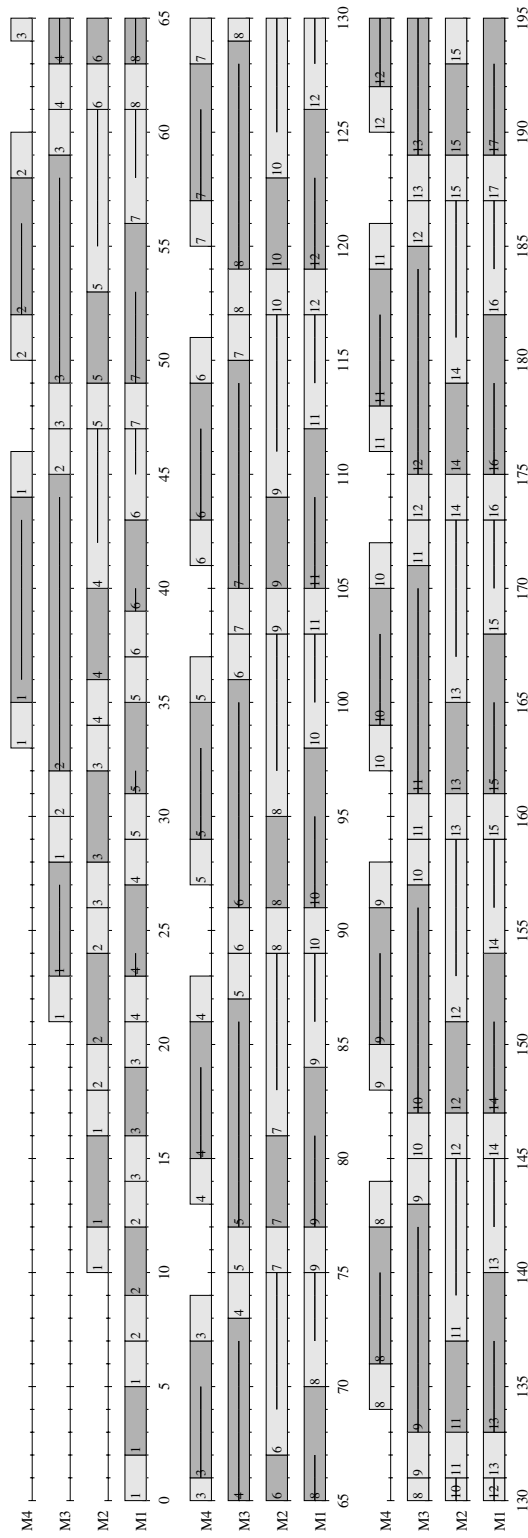


FIG. 5.20: Diagramme d'exécution avec la règle de tri LPT préemptive

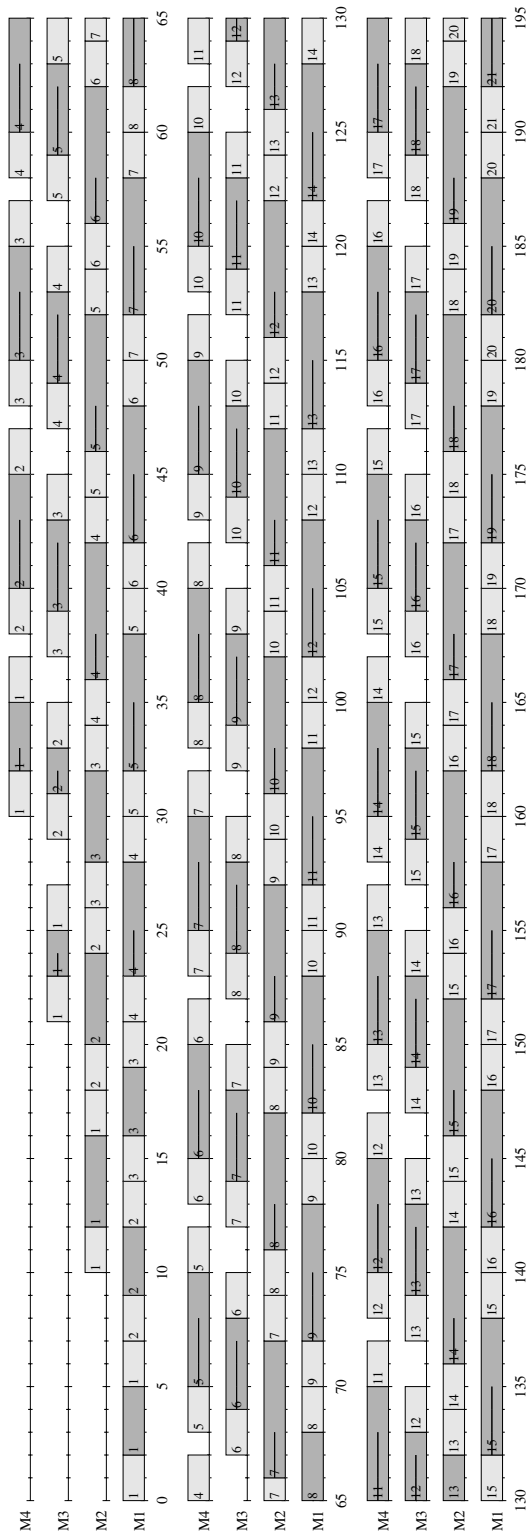


FIG. 5.21: Diagramme d'exécution avec la règle de tri Pull non-préemptive

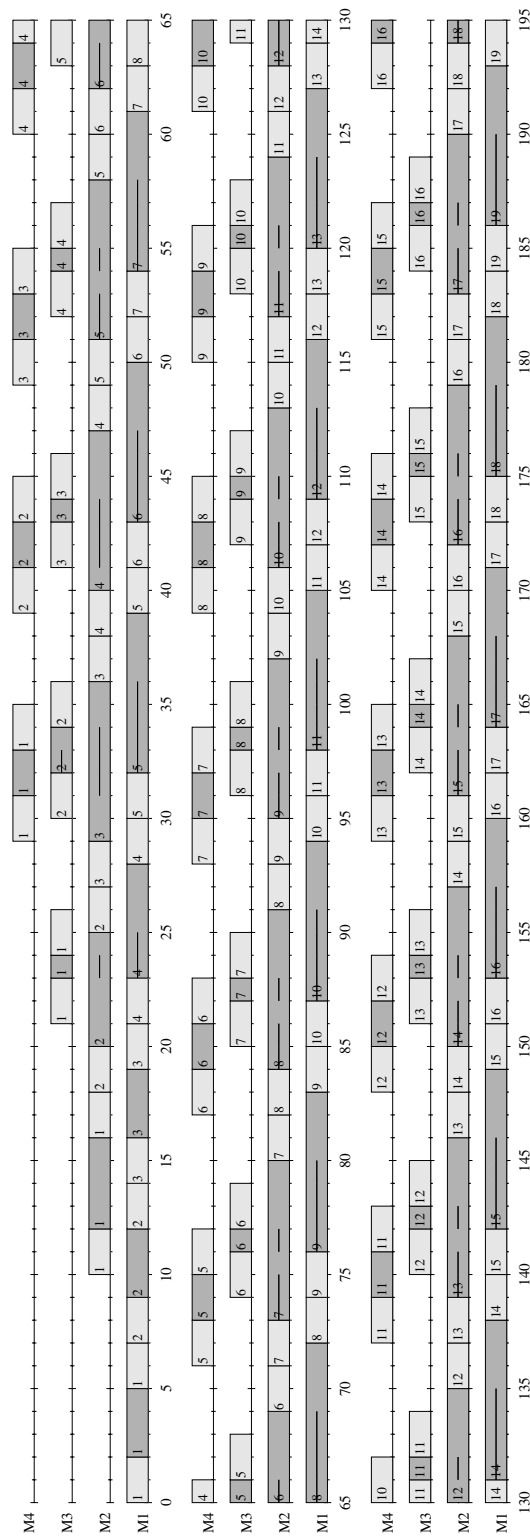


FIG. 5.22: Diagramme d'exécution avec la règle de tri Pull préemptive

Les rectangles colorés en gris clair correspondent aux opérations mécaniques exécutées localement par les machines de production. Le gris foncé est utilisé pour identifier les opérations distantes se rapportant aux traitements de messages de requête assurés par le contrôleur central. Les lignes horizontales barrant partiellement les opérations représentent les phases d'attente ou de suspension de leur exécution. Dans le cas d'une opération locale, une ligne marque une rétention d'une pièce sur une machine au terme du déroulement de son cycle opératoire. Lorsqu'il s'agit d'une opération distante, une ligne indique une attente avant le début du traitement ou une préemption du message de requête concerné au profit d'un autre plus prioritaire.

Règle de tri	τ	$\frac{\tau_{max}}{\tau}$ (%)
<i>FIFO</i>	10	100.00
<i>LIFO</i> non-préemptive	12.5	80.00
<i>LIFO</i> préemptive	12	83.33
<i>LPT</i> non-préemptive	14	71.43
<i>LPT</i> préemptive	14	71.43
<i>Pull</i> non-préemptive	10	100.00
<i>Pull</i> préemptive	11	90.09

TAB. 5.5: Comparaison des cadences obtenues

Les règles de tri *FIFO* et *Pull* non-préemptive permettent d'atteindre la borne sur la cadence de production et de satisfaire ainsi les conditions d'optimalité. La variante préemptive de la règle de tri *Pull* apporte une dégradation d'une unité de temps du délai τ d'inter-sorties de pièces par rapport à la borne τ . Mise à part la phase transitoire de début d'activité, les règles de tri *LPT* préemptive et non-préemptive produisent exactement les mêmes résultats et les diagrammes de Gantt qui en découlent sont identiques. Durant le régime stationnaire, le cas préemptif laisse d'ailleurs apparaître qu'aucune situation de préemption n'est rencontrée. Leur utilisation aboutit au plus faible niveau de productivité.

Le cas des règles de tri *LIFO* est particulier dans la mesure où elles provoquent un délai d'inter-sorties irrégulier. De plus, pour la variante préemptive, la date d'entrée en régime stationnaire est nettement retardée par rapport aux autres règles de tri. Afin de mieux étudier son comportement, nous avons procédé à des simulations spécifiques du système en prenant en compte une durée d'exécution plus importante. Nous pouvons

alors estimer à l'instant 81, lors de la fin de production de la quatrième pièce, la date d'établissement du régime stationnaire. La sortie des pièces du système respecte un cycle de six pièces selon des délais successifs de 6, 6, 22, 6, 26 puis 6 unités de temps. Le délai moyen d'inter-sorties de pièces est donc de 12 unités de temps. La règle de tri *LIFO* non-préemptive impose une sortie des pièces par cycles de deux pièces selon des délais de 14 et 11 unités de temps. Le délai d'inter-sorties de pièces obtenu avec la règle de tri *LIFO* non-préemptive est alors de 12.5 unités de temps et le régime stationnaire semble s'établir au terme de la fin de production de la deuxième pièce et de son relâchement par la quatrième machine.

Dans les cas simulés, il est intéressant de noter que l'utilisation du mécanisme de préemption ne permet que rarement de produire de meilleurs résultats sur la cadence de production en comparaison des variantes non-préemptives. Seul le cas des règles *LIFO* montre l'intérêt d'utiliser le mécanisme de préemption. Nous relevons également que le degré de productivité $\frac{\tau_{max}}{\tau}$ donné par rapport à la cadence maximale varie entre 71.43 % et 100 % selon la règle de tri utilisée.

5.6.3 Étude particulière de la règle de tri *FIFO*

Les résultats d'exécution présentés semblent montrer que l'utilisation de la règle de tri *FIFO* permet d'atteindre de manière quasi-systématique la borne τ_{max} sur le délai d'inter-sorties de travaux. Néanmoins, pour quelques rares cas issus de la simulation de systèmes de production existants, la règle de tri *FIFO* ne rend pas possible l'exécution du système flow-shop contrôlé dans des conditions optimales.

À travers un exemple particulier, il nous paraît alors nécessaire de confirmer, dans un premier temps, que l'optimalité de la cadence de production des systèmes flow-shop contrôlés n'est pas toujours garantie lorsque la règle de tri *FIFO* est mise en œuvre pour organiser l'activité du contrôleur central. De plus, nous montrons que son utilisation peut engendrer une dégradation significative du degré de productivité.

Pour étudier le cas particulier de la règle de tri *FIFO*, nous nous basons sur une configuration voisine de celle retenue dans le contexte de la simulation des systèmes existants en paramétrant une gamme opératoire équitablement répartie sur quatre machines de production. Un ensemble de cinq opérations est ainsi affecté à chacune des machines dont deux représentent des traitements de message de requête assurés par le contrôleur

central. Le tableau 5.6 détaille les données de paramétrage de la gamme opératoire que nous utilisons pour mettre en évidence la dégradation de productivité que peut provoquer l'utilisation de la règle de tri *FIFO*. Le délai $\psi_{i-1,i}$ de transfert des pièces et la capacité d'encours $\varpi_{i-1,i}$ entre deux machines successives sont respectivement fixés de manière homogène pour l'ensemble du système à deux unités de temps et trois pièces.

Opérations		M_i	θ_i	$\bar{\theta}_i$	τ_{max}
$\theta_{1,1}$	4	1	21	7	28
$\theta_{1,2}^*$	4				
$\theta_{1,3}$	9				
$\theta_{1,4}^*$	3				
$\theta_{1,5}$	1				
$\theta_{2,1}$	4	2	27	1	
$\theta_{2,2}^*$	4				
$\theta_{2,3}$	15				
$\theta_{2,4}^*$	3				
$\theta_{2,5}$	1				
$\theta_{3,1}$	4	3	24	4	
$\theta_{3,2}^*$	4				
$\theta_{3,3}$	12				
$\theta_{3,4}^*$	3				
$\theta_{3,5}$	1				
$\theta_{4,1}$	4	4	26	2	
$\theta_{4,2}^*$	4				
$\theta_{4,3}$	14				
$\theta_{4,4}^*$	3				
$\theta_{4,5}$	1				

TAB. 5.6: Gamme opératoire pour un cas de non-optimalité de la règle de tri *FIFO*

En raison de la durée de rétention des pièces sur les différentes machines du système flow-shop configuré et de la date tardive d'établissement du régime stationnaire, nous avons opté pour une durée simulée importante de la production. La figure 5.23 montre le diagramme de Gantt résultant de la simulation. Le régime stationnaire semble s'établir aux environs de l'instant 191, au moment du relâchement de la troisième pièce par la dernière machine. Aussi, nous limitons la présentation des résultats d'exécution à une durée de production simulée durant 390 unités de temps.

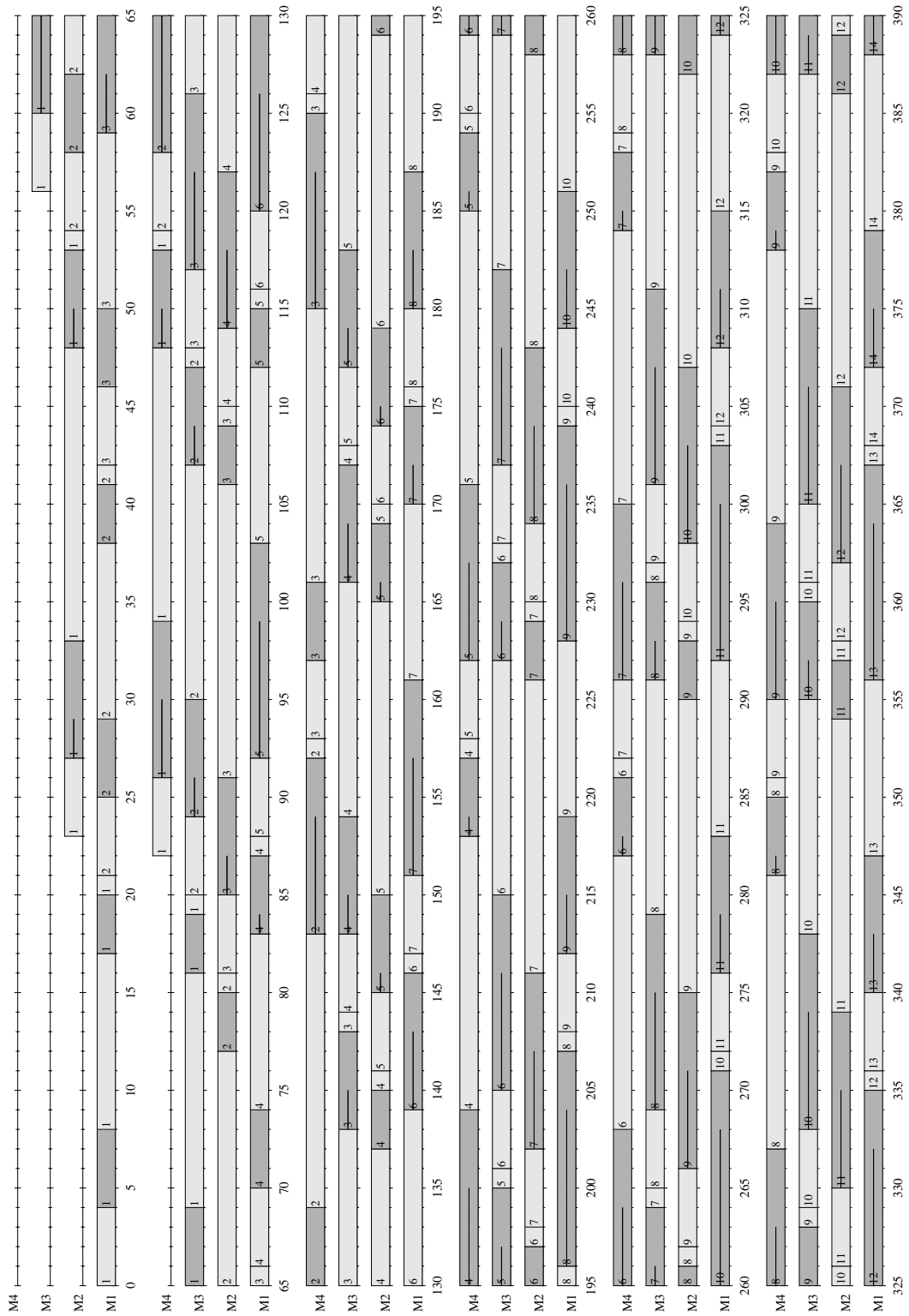


FIG. 5.23: Diagramme de démonstration de non-optimalité de la règle de tri FIFO

Selon le paramétrage du système, la borne sur le délai τ_{max} d'inter-sorties de pièces s'établit à 28 unités de temps par la charge du contrôleur central. Cependant, la consultation du diagramme de Gantt permet de constater que le délai τ d'inter-sorties de pièces en régime stationnaire reste limité à 32 unités de temps. La dégradation par rapport à la borne τ_{max} est alors de quatre unités de temps et le degré de productivité $\frac{\tau_{max}}{\tau}$ s'évalue à 87.5 % seulement. Nous constatons en outre que les machines de production sont en permanence occupées et qu'elles ne marquent jamais de temps de latence entre les prises en charge de deux pièces successives.

5.6.4 Évaluation d'une configuration bornant la cadence de production par la machine la plus lente

Les premières simulations complémentaires ont été menées en prenant en compte une gamme opératoire positionnant le contrôleur central de telle façon qu'il constitue la ressource critique du système étudié. Afin de les étendre, nous proposons d'évaluer une configuration définie pour que le goulet d'étranglement soit situé au niveau de la machine la plus lente et que la borne τ_{max} sur le délai d'inter-sorties de pièces soit ainsi d'ordre mécanique. Nous avons donc :

$$\sum_{M_i \in \mathcal{M}} \theta_i^* < \max_{M_i \in \mathcal{M}} \{\theta_i\} \quad (5.6)$$

Notre objectif est, pour ce cas particulier, d'évaluer les possibilités d'existence d'une dégradation du degré de productivité lorsque la borne mécanique prévaut sur la borne de contrôle. Cette situation n'a, en effet, pas été mise en évidence lors des simulations initiales basées sur les systèmes existants. Les règles de tri que nous avons retenues sont *FIFO* et *LPT* non-préemptive pour leur écart de comportement lors des premières simulations complémentaires.

Afin de faire prévaloir la borne mécanique, nous adaptons la gamme opératoire précédemment donnée par le tableau 5.4 en augmentant la durée des opérations $\theta_{2,1}$ et $\theta_{2,3}$ respectivement à quatre et trois unités de temps. La borne mécanique est ainsi portée à onze unités de temps par la seconde machine alors que la borne de contrôle est maintenue à dix unités de temps. Les figures 5.24 et 5.25 montrent les diagrammes de Gantt résultant de l'exécution.

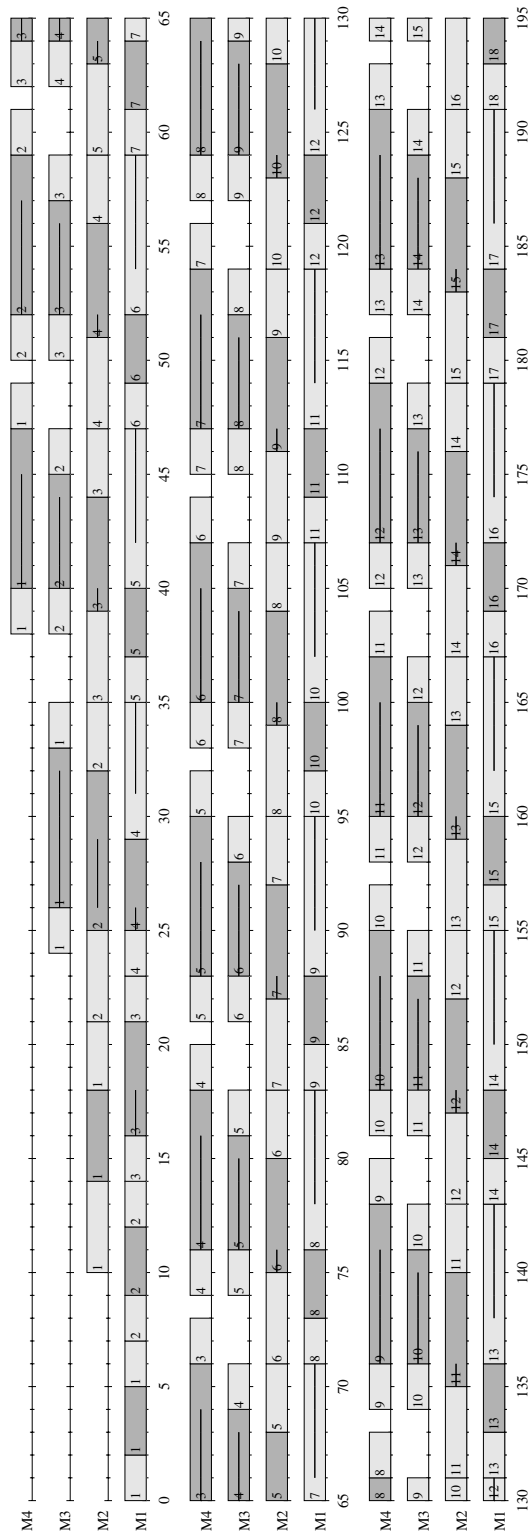


FIG. 5.24: Diagramme d'exécution spécifique avec la règle de tri FIFO

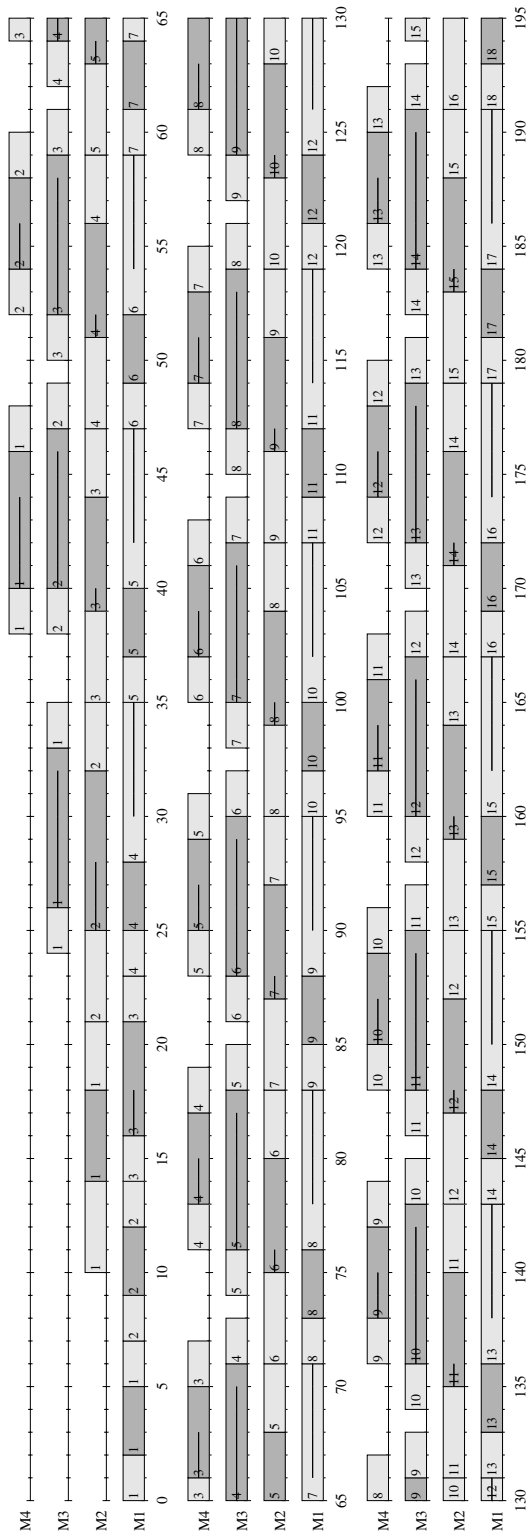


FIG. 5.25: Diagramme d'exécution spécifique avec la règle de tri LPT non-préemptive

Les effets sur la cadence de production sont identiques. Les règles de tri *FIFO* et *LPT* non-préemptive ne permettent pas d'atteindre la borne τ_{max} sur le délai d'inter-sorties de pièces en permettant au système de produire une pièce toutes les onze unités de temps. Le délai τ d'inter-sorties de pièces résultant de la simulation est établi à douze unités de temps. Il montre une dégradation de la cadence de production. Le degré de productivité $\frac{\tau_{max}}{\tau}$ se situe alors à 91.67 %.

L'enchaînement des traitements successifs des pièces sur les différentes machines de production sont proches. Les dates de prise en charge et de relâchement des pièces sont sensiblement les mêmes sur les deux diagrammes de Gantt.

5.7 Conclusion

Le but de la phase de simulation présentée dans ce chapitre était de mesurer l'impact du contrôleur central sur le niveau de performances des systèmes flow-shop contrôlés considérés dans nos travaux. Les résultats obtenus permettent de révéler des propriétés comportementales importantes des procédés concernés. Ils mettent avant tout en évidence l'existence d'une borne de contrôle limitant, dans la majorité des cas, la cadence de production à une valeur plus faible que celle établie par la machine la plus lente. Nous avons alors exhiber une borne τ_{max} sur le délai d'inter-sorties de pièces quelle que soit la ressource critique du système.

L'étude comparative de diverses règles de tri utilisées pour organiser l'activité du contrôleur central montre qu'elles produisent des effets différents et parfois éloignés sur le délai τ d'inter-sorties de pièces. Elle a également permis de révéler que, pour les cas évalués, une dégradation du degré de productivité peut être constatée même lorsque la borne τ_{max} est fixée par la machine la plus lente. Aucune règle de tri ne produit de manière systématique des résultats optimaux en garantissant l'atteinte de la borne sur le délai d'inter-sorties de pièces, quelle que soit la configuration du système flow-shop considéré.

Le choix de la règle de tri apparaît alors comme une décision importante à considérer dans l'optimisation de la cadence de production, et donc du makespan. L'objectif de nos travaux consiste désormais à se rapprocher au plus près, voire d'atteindre, la borne de contrôle lorsque celle-ci prévaut sur la borne mécanique. Dans le cas contraire, il s'agit de garantir que la cadence de production s'établisse à la borne mécanique.

Partie III

Résolution de la problématique

Chapitre 6

Atteinte du délai optimal d'inter-sorties de pièces

Sommaire

6.1	Sélection des facteurs d'optimalité	152
6.1.1	Respect de la périodicité d'activité des machines de production et du contrôleur central	153
6.1.2	Gestion de l'accès au contrôleur central	154
6.2	Méthode de synchronisation des machines de production	154
6.2.1	Principe général de la méthode	155
6.2.2	Représentation et permutation de la séquence opératoire d'une machine de production	156
6.2.3	Évaluation des combinaisons d'une gamme opératoire	162
6.2.4	Répartition de la fonction de coordination	167
6.2.5	Critères complémentaires pour la sélection d'une combinaison d'une gamme opératoire	168
6.2.6	Principales conséquences de la synchronisation des machines de production	169
6.3	Conclusion	170

Lors de la conception d'un système flow-shop contrôlé, il est nécessaire de s'assurer que le procédé sera en mesure de produire à la cadence maximale en respectant un délai moyen optimal τ_{max} d'inter-sorties de pièces. Dans le cas contraire, il paraît intéressant de connaître a priori l'écart à l'optimalité en évaluant le niveau de performances atteignable.

L'activité de traitement des messages de requête par le contrôleur nécessite d'être optimisée afin de pénaliser au minimum la cadence de production du système contrôlé. Les messages de réponse doivent alors être délivrés aux machines dans des délais bornés en fonction de la dynamique du procédé. Il est donc nécessaire d'évaluer, pour chaque message de requête envoyé au contrôleur, une échéance de fin de traitement et d'envoi du message de réponse correspondant qui satisfassent les conditions d'exécution optimale du système contrôlé.

Les résultats obtenus lors de la phase de simulation ont montré que le choix de la règle de tri influait significativement sur le degré de productivité des systèmes. Dans un premier temps, il convient d'évaluer les facteurs influant directement sur le niveau de performances des systèmes et notamment sur leur capacité à produire à la cadence optimale. Les indications obtenues ont pour but de permettre la vérification de l'existence d'une possibilité pour le système d'atteindre le délai moyen optimal τ_{max} d'inter-sorties de pièces en analysant les accès concurrents de messages de requête envoyés au contrôleur par les machines de production. Ensuite, nous proposons une méthode de synchronisation du cycle opératoire des machines de production permettant d'aboutir à une maîtrise des instants auxquels sont envoyés les messages de requête au contrôleur central et ainsi d'éviter des concurrences d'accès pénalisantes pour la cadence de production. Elle ne représente cependant qu'une approche statique de la réponse à la problématique posée.

6.1 Sélection des facteurs d'optimalité

Plusieurs facteurs entrent en ligne de compte dans le niveau de performances d'un système flow-shop contrôlé et, plus précisément, dans sa capacité à approcher ou atteindre la borne τ_{max} sur le délai d'inter-sorties de pièces. Dans le cadre de nos travaux, les facteurs importants correspondent aux critères que doit satisfaire au mieux la règle de tri utilisée au niveau du contrôleur. Ils concernent essentiellement l'échéance de fin de traitement des messages de requête et la gestion des accès au contrôleur central. Le but

recherché étant de permettre au système de respecter au mieux les dates optimales de démarrage et de fin d'exécution du cycle opératoire des machines de production pour chacune des pièces à produire.

6.1.1 Respect de la périodicité d'activité des machines de production et du contrôleur central

Sur un plan général et de manière idéale, lorsque l'activité d'un système flow-shop se déroule en régime stationnaire, les machines de production déroulent leur cycle opératoire à intervalles réguliers. Le délai séparant les dates de début de traitement sur une même machine de deux pièces consécutives est égal au délai d'inter-sorties de pièces. Nous pouvons alors admettre que l'ensemble des machines d'un système flow-shop est soumis à une période commune d'activité donnée par le délai τ d'inter-sorties de pièces et que celle-ci est optimale lorsqu'elle est égale à la borne τ_{max} sur la cadence de production. Nous désignons cette période par le terme *période fondamentale* du système. La figure 6.1 résume la périodicité commune à toutes les machines de production d'un système flow-shop.

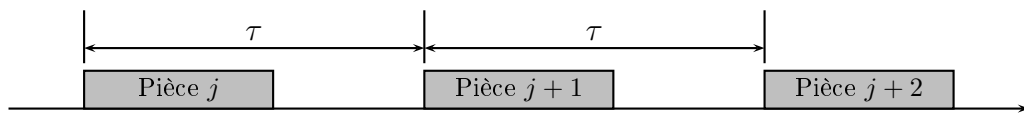


FIG. 6.1: Périodicité commune des machines de production

Par conséquent, lorsqu'il s'agit d'un système flow-shop non-contrôlé, il est important de ne pas ralentir la machine la plus lente qui fixe la cadence de production. Ce premier postulat conduit à la déduction d'un autre permettant d'avancer l'idée que le déroulement du cycle opératoire d'une machine M_i peut être suspendu durant une certaine durée dans la limite de sa latence d'activité $\bar{\theta}_i$ sans pénaliser le délai τ d'inter-sorties de pièces. La périodicité de la machine est ainsi respectée.

Dans le cas d'un système flow-shop mettant en œuvre un contrôleur central, nous relevons également qu'il est possible de geler l'activité de toutes les machines de production pendant une durée maximale égale à leur latence. Ce ralentissement limité dans le temps garantit néanmoins que le délai d'inter-sorties de pièces est optimal et permet au procédé d'atteindre la borne sur la cadence de production, soit $\tau = \tau_{max}$.

Le contrôleur central est une ressource à part entière d'un système flow-shop contrôlé. Identiquement aux machines de production, son activité doit également respecter la période fondamentale pour ne pas pénaliser la cadence du procédé. Notre problème d'ordonnancement s'inscrit alors dans la recherche d'un enchaînement réalisable au cours de la période fondamentale de toutes les unités de temps de la gamme opératoire associées à un traitement de message de requête.

6.1.2 Gestion de l'accès au contrôleur central

Le facteur le plus évident à dégager s'applique aux accès concurrents au contrôleur central. En effet, la borne τ_{max} sur le délai d'inter-sorties de pièces est systématiquement accessible lorsque les messages de requête sont traités sans attente. Cette situation représente évidemment le cas idéal. Cependant, le contrôleur constitue une ressource disjonctive ne permettant que le traitement unitaire des messages de requête. Le premier rôle de la règle de tri est alors de sélectionner le message à traiter le plus urgemment lorsque le contrôleur redevient disponible.

La règle de tri utilisée peut aussi, dans une certaine mesure, permettre un asservissement du système flow-shop contrôlé de telle façon que les machines de production soient coordonnées et synchronisées afin de ne solliciter le contrôleur qu'à des instants bien définis. En procédant ainsi, il s'agit de contribuer à ce que les messages de requête ne fassent pas l'objet d'une attente trop longue avant leur prise en charge ou de préemptions trop fréquentes qui ne permettraient plus aux machines de respecter la périodicité du système.

6.2 Méthode de synchronisation des machines de production

Bien qu'étant une ressource à part entière du système, le contrôleur central se distingue des machines de production par la nature des opérations exécutées mais aussi par l'inexistence de contraintes de précedence/antériorité dans l'ordre de traitement des unités de temps de la séquence opératoire qui lui est affectée. La seule restriction, quant à la cadence de production, concerne le respect de la fréquence fondamentale du système.

L'exécution du cycle opératoire de chacune des machines de production s'effectue, dans un contexte courant, en respectant strictement une période τ commune à l'ensemble du système. Cependant, la date au plus tôt de début de première exécution, et donc d'entrée en production, est propre à chaque machine. Elle dépend directement des durées opératoires des machines implantées en amont sur le système et des délais de transfert jusqu'à la machine concernée. Nous proposons, par une méthode spécifique, de paramétrer la synchronisation des machines de production en influant sur leur date d'entrée en production et en utilisant leur délai de latence pour maîtriser la simultanéité des accès au contrôleur central.

6.2.1 Principe général de la méthode

Le principe de la méthode repose sur une consommation des unités de temps de latence $\bar{\theta}_i$ disponibles pour retarder ou suspendre le traitement des messages de requête émis par une machine de production M_i sans violer une échéance fixée par rapport à la date au plus tôt de prise en charge de la pièce suivante. Il prend également en considération la possibilité de repousser la date d'entrée en production des machines afin de mieux les synchroniser de telle sorte que la cadence de production soit maximisée.

Il s'agit avant tout de vérifier une condition nécessaire quant à la charge du contrôleur central engendrée par le traitement des messages de requête au cours de chaque période de τ_{max} unités de temps. L'aspect principal de la méthode réside dans la recherche de l'existence d'au moins une combinaison particulière, que nous qualifions de *combinaison optimale*, des séquences opératoires de chacune des machines du système. Une combinaison optimale exclue la possibilité d'avoir plusieurs messages de requête soumis de façon concurrente au contrôleur central imposant la nécessité de traitement durant la même unité de temps.

La figure 6.2 montre, à travers un exemple, le principe de synchronisation des machines. Les deux machines M_1 et M_2 représentées sont respectivement affectées d'une séquence opératoire constituée de trois et quatre opérations. Les unités de temps relatives à la gamme opératoire sont numérotées dans l'ordre chronologique de leur exécution pour une pièce donnée. Celles relevant d'un traitement distant assuré par le contrôleur sont grisées. La borne τ_{max} sur le délai d'inter-sorties de pièces est fixée à quatre unités de temps conjointement par la machine M_2 et le contrôleur central. La machine M_1 fait

l'objet d'une suspension du déroulement de son cycle opératoire au terme de l'exécution de la première unité de temps. La seconde unité de temps de sa séquence opératoire correspond au traitement d'un message de requête par le contrôleur. Son exécution est retardée au profit de la machine M_2 . Dans le cas présenté, la périodicité d'activité de toutes les ressources, machines et contrôleur, est respectée et s'établit par rapport à la période fondamentale du système permettant ainsi d'atteindre une cadence de production optimale.

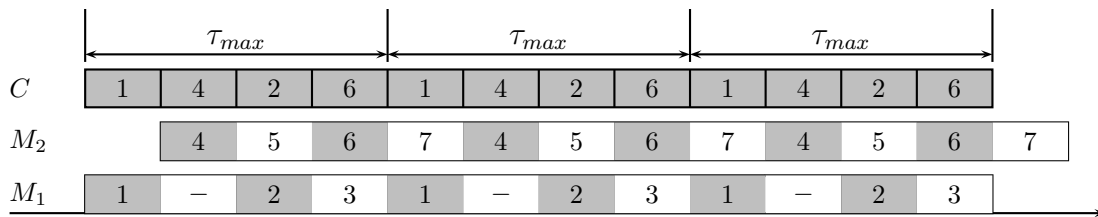


FIG. 6.2: Synchronisation des machines de production

Lorsqu'une combinaison optimale des séquences opératoires existe, elle permet au contrôleur central de ne pas pénaliser le délai d'inter-fins de travaux en effectuant un traitement synchronisé des messages de requête. Chacune des machines peut alors respecter la périodicité du système en démarrant une nouvelle exécution de son cycle opératoire toutes les τ_{max} unités de temps. Lorsqu'aucune combinaison optimale ne peut être trouvée, la méthode permet néanmoins d'établir une combinaison qui minimise le taux de dégradation de la productivité du système.

6.2.2 Représentation et permutation de la séquence opératoire d'une machine de production

Les opérations de la gamme opératoire nécessaire à la production d'une pièce sont réparties sur les différentes machines selon un ordre figé. Leur délai d'exécution effectif est supposé connu et constant. Il est alors possible de représenter la séquence opératoire, que nous notons η_i , d'une machine de production M_i sous la forme d'un découpage en τ_{max} unités de temps, correspondant à la période à respecter par l'ensemble des ressources du système, en distinguant par un marquage numérique les opérations exécutées localement de celles confiées au contrôleur central.

Le tableau 6.1 présente les données d'un exemple de système flow-shop constitué de trois machines destinées à la réalisation de pièces nécessitant l'exécution de onze opérations. Les unités de temps marquées par un tiret indiquent que la machine M_i concernée dispose d'une latence de durée $\overline{\theta}_i$ entre deux pièces successives fixée par rapport à la période fondamentale τ_{max} . Les unités de temps associées au chiffre 0 correspondent s'appliquent aux opérations mécaniques exécutées localement par les machines alors que le chiffre 1 fait état du traitement distant d'un message de requête assuré par le contrôleur. Nous désignons chaque séquence opératoire η_i ainsi représentée par le terme de *séquence initiale*. La séquence initiale η_i d'une machine M_i est ainsi caractérisée par un triplet $(\theta_i, \theta_i^*, \overline{\theta}_i)$.

Opérations		M_i	θ_i	θ_i^*	$\overline{\theta}_i$	η_i	τ_{max}						
$\theta_{1,1}^*$	1	1	6	3	0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td> </tr> </table>	1	0	0	1	0	1	6
1	0						0	1	0	1			
$\theta_{1,2}$	2												
$\theta_{1,3}^*$	1												
$\theta_{1,4}$	1												
$\theta_{1,5}^*$	1												
$\theta_{2,1}$	1	2	4	2	2	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0</td><td>1</td><td>1</td><td>0</td><td>-</td><td>-</td> </tr> </table>	0	1	1	0	-	-	
0	1						1	0	-	-			
$\theta_{2,2}^*$	2												
$\theta_{2,3}$	1												
$\theta_{3,1}$	1	3	3	1	3	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0</td><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td> </tr> </table>	0	1	0	-	-	-	
0	1						0	-	-	-			
$\theta_{3,2}^*$	1												
$\theta_{3,3}$	1												

TAB. 6.1: Représentation d'une gamme opératoire

Les conditions de synchronisation de l'activité des machines du système s'appuient sur la position des unités de temps liées aux messages de requête pour aboutir à l'exclusion des conflits d'accès au contrôleur central durant chaque unité de temps de la période fondamentale τ_{max} visée. Le but est alors d'établir une combinaison de séquences opératoires déclinées par permutation à partir de la séquence initiale η_i qui satisfasse cette synchronisation. Chaque séquence initiale peut en effet être permutée par rotation pour exprimer l'activité d'une machine de production M_i sur la période fondamentale τ_{max} du système dans des conditions d'exécution optimales. Il s'agit de définir un ordre d'exécution des unités de temps à traiter par le contrôleur au cours d'une période τ_{max} sans application d'une contrainte de précedence/antériorité.

Les séquences opératoires pouvant être déclinées par permutation à partir de la séquence initiale de la machine de production doivent tenir compte des conditions d'évolution des systèmes flow-shop considérés et de la règle de précedence/antériorité qui s'applique au sein du jeu d'opérations. L'ordre d'exécution des opérations pour un travail donné ne pouvant pas être modifié pour les machines de production.

Dans le cas où la séquence initiale n'intègre pas d'unités de temps de latence, notamment lorsque la machine concernée constitue le goulet d'étranglement du système, nous avons $\theta_i = \tau_{max}$. Le nombre de séquences déclinées, que nous notons \mathcal{C}_i , est alors de θ_i et s'obtient simplement par rotations successives :

$$\forall \theta_i \geq \theta_i^*, \forall \bar{\theta}_i = 0, \forall \theta_i^* \geq 1 : \mathcal{C}_i = \theta_i \quad (6.1)$$

Nous exprimons par \mathcal{S}_i l'ensemble des déclinaisons de la séquence opératoire affectée à la machine de production M_i sur une durée équivalente à la période fondamentale τ_{max} . Cet ensemble inclut les séquences initiale et déclinées. Ainsi, pour une séquence initiale $\eta_i = |110|$ avec $\tau_{max} = 3$, il existe un ensemble \mathcal{S}_i de trois séquences opératoires, soit $\mathcal{S}_i = \{|110|; |011|; |101|\}$.

Lorsque pour une machine M_i , la séquence initiale η_i intègre au moins une latence $\bar{\theta}_i$, nous avons $\theta_i < \tau_{max}$. La déclinaison de l'ensemble des séquences opératoires d'une machine s'effectue par combinaisons en tenant compte de la possibilité de placement exclusif des unités de temps de latence en fin de séquence ou devant une unité de temps liée à un traitement de message de requête par le contrôleur. Le déplacement d'une unité de temps de latence représente ainsi l'insertion d'une attente avant le début du traitement ou d'une préemption au profit d'un message de requête présentant une plus forte priorité. À titre d'exemple, la séquence initiale $\eta_i = |101-|$ d'une machine M_i avec $\tau_{max} = 4$, se décline en un ensemble \mathcal{S}_i de neuf séquences opératoires. Nous obtenons alors $\mathcal{S}_i = \{|101-|; |10-1|; |-101|; |110-|; |1-10|; |-110|; |011-|; |01-1|; |0-11|\}$.

Une décomposition s'effectue en deux étapes à partir de la séquence initiale η_i d'une machine M_i . Il s'agit, dans un premier temps, de déterminer les θ_i séquences déclinées par simple rotation de la séquence opératoire correspondant aux θ_i unités de temps d'activité effective sans tenir compte de la latence $\bar{\theta}_i$. La seconde étape consiste à déplacer successivement les $\bar{\theta}_i$ unités de temps de latence devant chacune des θ_i^* unités de temps correspondant à un traitement de message de requête par le contrôleur. La

décomposition 6.2 résume le principe de déclinaison de séquences opératoires à partir de la séquence initiale η_i .

$$\left| 1 \ 0 \ 1 \ - \right| \Rightarrow \left\{ \begin{array}{l} \left| 1 \ 0 \ 1 \ - \right| \Rightarrow \begin{vmatrix} 1 & 0 & 1 & - \\ 1 & 0 & - & 1 \\ - & 1 & 0 & 1 \end{vmatrix} \\ \\ \left| 1 \ 1 \ 0 \ - \right| \Rightarrow \begin{vmatrix} 1 & 1 & 0 & - \\ 1 & - & 1 & 0 \\ - & 1 & 1 & 0 \end{vmatrix} \\ \\ \left| 0 \ 1 \ 1 \ - \right| \Rightarrow \begin{vmatrix} 0 & 1 & 1 & - \\ 0 & 1 & - & 1 \\ 0 & - & 1 & 1 \end{vmatrix} \end{array} \right. \quad (6.2)$$

Le nombre de positions autorisées pour les unités de temps de latence correspond à la somme des temps de traitement de message de requête et de latence, soit $\theta_i^* + \bar{\theta}_i$. Il s'agit alors d'établir les possibilités de placement de $\bar{\theta}_i$ unités de temps de latence parmi $\theta_i^* + \bar{\theta}_i$ positions. Le nombre $\mathcal{C}_{\theta_i^* + \bar{\theta}_i}^{\bar{\theta}_i}$ de déclinaisons possibles pour chaque permutation obtenue par rotation d'une séquence opératoire initiale donnée s'évalue donc par un calcul de combinaisons sans répétition :

$$\forall \bar{\theta}_i \geq 1, \forall \theta_i^* \geq 1 : \mathcal{C}_{\theta_i^* + \bar{\theta}_i}^{\bar{\theta}_i} = \frac{(\theta_i^* + \bar{\theta}_i)!}{((\theta_i^* + \bar{\theta}_i) - \bar{\theta}_i)! \times \bar{\theta}_i!}$$

$$\mathcal{C}_{\theta_i^* + \bar{\theta}_i}^{\bar{\theta}_i} = \frac{(\theta_i^* + \bar{\theta}_i)!}{\theta_i^*! \times \bar{\theta}_i!} \quad (6.3)$$

Pour obtenir le nombre global \mathcal{C}_i de combinaisons relatives à l'ensemble \mathcal{S}_i des séquences opératoires d'une machine M_i , il faut étendre l'équation 6.3 en prenant en compte le nombre θ_i de permutations déclinaibles par rotation à partir de la séquence initiale considérée sans temps de latence :

$$\forall \theta_i \geq \theta_i^*, \forall \bar{\theta}_i \geq 1, \forall \theta_i^* \geq 1 : \mathcal{C}_i = \frac{(\theta_i^* + \bar{\theta}_i)!}{\theta_i^*! \times \bar{\theta}_i!} \times \theta_i \quad (6.4)$$

Dans le cas où le placement d'une unité de temps de latence devant une unité de temps liée à un traitement de message de requête par le contrôleur coïncide avec le début de séquence, il peut être assimilé à un retard appliqué au début d'exécution du cycle opératoire de la machine concernée. Dans ce contexte, nous devons prendre en compte la possibilité de différer le début d'exécution du cycle opératoire de toutes les machines et intégrer le fait qu'une unité de temps de latence puisse aussi être placée en tête de séquence devant une unité de temps relative à l'exécution d'une opération locale. Le nombre global \mathcal{C}_i de combinaisons relatives à l'ensemble \mathcal{S}_i des séquences opératoires d'une machine M_i , donné par l'équation 6.4, doit donc être complété par l'addition du nombre d'unités de temps correspondant aux opérations locales, soit $\theta_i - \theta_i^*$:

$$\forall \theta_i \geq \theta_i^*, \forall \bar{\theta}_i \geq 1, \forall \theta_i^* \geq 1 : \mathcal{C}_i = \left(\frac{(\theta_i^* + \bar{\theta}_i)!}{\theta_i^*! \times \bar{\theta}_i!} \times \theta_i \right) + (\theta_i - \theta_i^*) \quad (6.5)$$

Le calcul de déclinaisons de la séquence opératoire initiale η_i successivement présenté par les équations 6.3, 6.4 et 6.5 n'est possible que sous certaines conditions. La machine M_i doit disposer d'au moins une unité de temps de latence à consommer entre le traitement de deux pièces consécutives et que le nombre d'unités de temps liées à un traitement de message de requête ne soit pas nul. En effet, lorsque le nombre d'unité de temps de latence est nul, le calcul se limite à θ_i combinaisons pour une machine M_i et lorsque le nombre d'unité de temps de traitement de messages de requête est nul, la séquence opératoire de la machine n'entre pas en ligne de compte dans l'organisation de l'activité du contrôleur central. Le nombre \mathcal{C}_i de combinaisons pour une machine M_i donné par l'équation 6.5 s'applique ainsi à une machine de production ne constituant pas une ressource critique du système :

$$\mathcal{C}_i = f(\forall M_i \in \mathcal{M} : \theta_i, \theta_i^*, \bar{\theta}_i) \quad (6.6)$$

La figure 6.3 montre l'accroissement du nombre \mathcal{C}_i de combinaisons possibles pour une machine de production M_i ne constituant pas une ressource critique du système flow-shop contrôlé. Le graphique supérieur est établi à partir des données calculées en fonction du nombre $\bar{\theta}_i$ d'unités de temps de latence dont dispose la machine et en considérant que le traitement distant de messages de requête se limite à seule unité de temps. Le graphique inférieur est réciproquement basé sur l'évolution du nombre θ_i^*

d'unités de temps consacrées au traitement de messages de requête par le contrôleur central en ne prenant en compte qu'une unité de temps de latence. Il s'agit de montrer l'influence sur le nombre \mathcal{C}_i de combinaisons engendrée par l'évolution d'une variable donnée indépendamment des autres. Nous avons alors $\bar{\theta}_i = 1$ à 5 et $\theta_i^* = 1$ dans le premier cas puis $\theta_i^* = 1$ à 5 et $\bar{\theta}_i = 1$ dans le second.

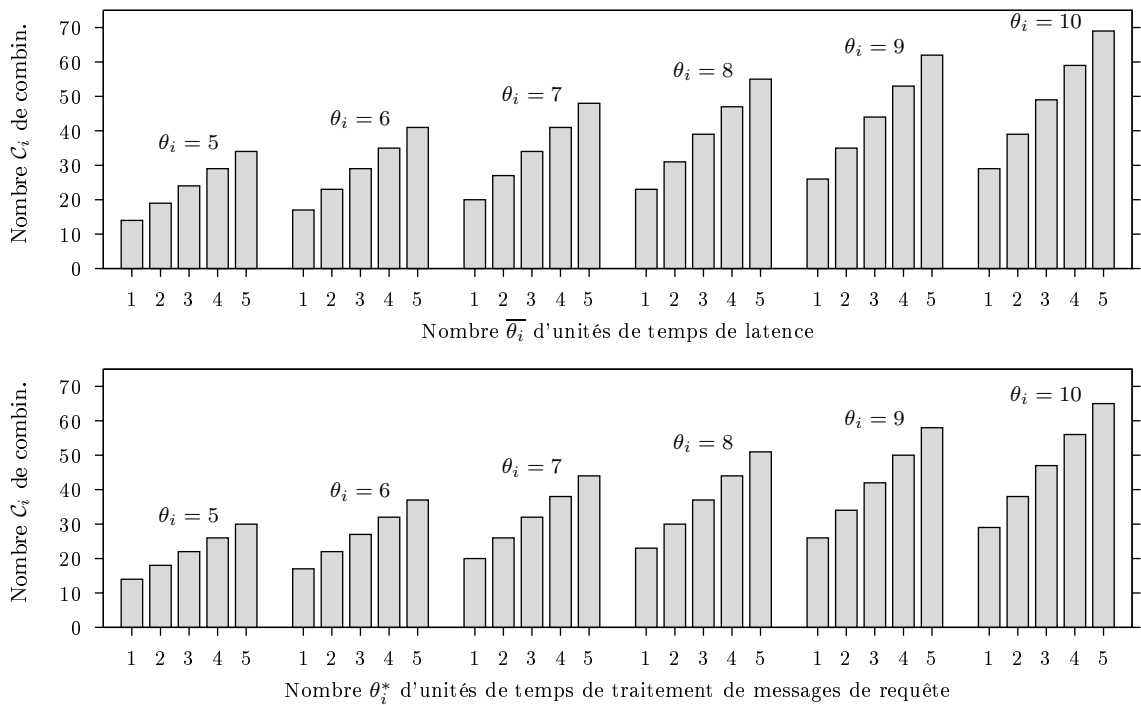


FIG. 6.3: Visualisation du nombre de combinaisons pour une machine non-critique

Les deux graphiques présentent des aspects très similaires. Le nombre de combinaisons croît de façon linéaire lorsque le nombre d'unités de temps de latence ou de traitement distant de messages de requête augmente. Il progresse également de manière identique en fonction de la durée θ_i d'exécution de la séquence opératoire affectée à la machine M_i évaluée entre cinq et dix unités de temps.

Le nombre \mathcal{C}_i constitue l'espace maximal de combinaisons existant pour une machine M_i donnée. Il peut, dans certains cas, être réduit en fonction de la constitution de la séquence opératoire initiale η_i et de mode d'alternance des unités de temps liées à un traitement local ou distant. En effet, lorsque que la séquence opératoire initiale est constituée d'une succession d'un même enchaînement d'unités de temps, certaines séquences

opératoires déclinées sont identiques. Une réduction de l'espace de combinaisons peut être notamment envisagé lorsque la séquence opératoire de la machine concernée n'intègre pas d'unités de temps de latence. Le nombre \mathcal{C}_i se détermine après identification de la sous-séquence qui se répète. La décomposition 6.7 montre un exemple de séquence opératoire initiale aboutissant à une limitation du nombre de déclinaisons et donc à une réduction de l'ensemble \mathcal{S}_i des séquences déclinées pour une machine M_i .

$$\underbrace{\left| \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right|}_{\eta_i} \Rightarrow \underbrace{\left| \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{array} \right|}_{\mathcal{S}_i \text{ avant réduction}} \Rightarrow \underbrace{\left| \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{array} \right|}_{\mathcal{S}_i \text{ après réduction}} \quad (6.7)$$

Dans l'exemple présenté, le nombre \mathcal{C}_i est réduit de six à trois séquences déclinées. La sous-séquence $|101|$ répétée dans la séquence opératoire initiale est en effet d'une durée de trois unités de temps. Elle conditionne la réduction du nombre de séquences déclinées.

6.2.3 Évaluation des combinaisons d'une gamme opératoire

Nous représentons une combinaison de la gamme opératoire de l'ensemble d'un système flow-shop contrôlé sous la forme d'un tableau de τ_{max} colonnes représentant les unités de temps et de k lignes correspondant aux machines. Son évaluation repose sur l'addition des valeurs d'une colonne. Les unités de temps de latence affectées d'un tiret sont considérées comme nulles et n'impactent pas le total de la colonne. Les résultats supérieurs à 1 indiquent une concurrence d'accès au contrôleur central et expriment l'invalidité potentielle de la combinaison dans un contexte d'optimalité des performances du système considéré.

Le tableau 6.2 s'appuie sur les données présentées dans le tableau 6.1 et montre le principe d'évaluation d'une combinaison. Il permet de mettre en évidence un conflit d'accès au contrôleur central lors de la seconde unité de temps de la période fondamentale τ_{max} du système. La séquence opératoire obtenue montre également que le contrôleur

dispose d'une unité de temps d'inactivité sur la période τ_{max} .

1	0	0	1	0	1
0	1	1	0	-	-
0	1	0	-	-	-
1	2	1	1	0	1

TAB. 6.2: *Évaluation d'une combinaison*

Alors qu'une combinaison est optimale dès lors qu'aucune somme de colonne n'est supérieure à 1, la pertinence d'une combinaison qui ne l'est pas peut néanmoins être estimée en fonction de son écart à l'optimalité et donc du nombre d'unités de temps supplémentaires ajoutées à la fréquence fondamentale τ_{max} du système établissant le délai d'inter-fins de travaux. À titre d'exemple, le résultat d'évaluation $|1\ 2\ 1\ 1\ 0\ 1|$ de la combinaison présentée par le tableau 6.2 indique que le délai d'inter-fins de travaux est porté à sept unités de temps bien que le niveau d'optimalité est établi à τ_{max} , soit six unités de temps. Ce qui implique une dégradation voisine de 17 % du degré de productivité du système flow-shop considéré.

Afin de calculer le nombre total $\mathcal{C}_{\mathcal{M}}$ de combinaisons existantes pour un ensemble \mathcal{M} des machines d'un système donné, nous devons distinguer les machines ne disposant pas d'un temps de latence de celles dont la durée θ_i de séquence opératoire est inférieure à la période fondamentale τ_{max} . Le mode d'évaluation du nombre de combinaisons, donné par les équations 6.1 et 6.5, est en effet différent selon les cas. Aussi, nous séparons les machines de l'ensemble \mathcal{M} en deux sous-ensembles \mathcal{M}^\bullet et \mathcal{M}° pour distinguer respectivement les machines les plus lentes du système dont le déroulement du cycle opératoire ne peut être suspendu de celles autorisant un gel de leur activité à certains moments. Nous avons alors $\forall M_i \in \mathcal{M}^\bullet : \theta_i = \tau_{max}, \bar{\theta}_i = 0$ et $\forall M_i \in \mathcal{M}^\circ : \theta_i < \tau_{max}, \bar{\theta}_i \geq 1$. Le calcul se décline sous la forme d'un produit basé sur la séquence opératoire affectée à chacune des machines du système :

$$\forall \theta_i \geq \theta_i^*, \forall \bar{\theta}_i \geq 1, \forall \theta_i^* \geq 1 : \mathcal{C}_{\mathcal{M}} = \prod_{M_i \in \mathcal{M}^\circ} \left(\left(\frac{(\theta_i^* + \bar{\theta}_i)!}{\theta_i^*! \times \bar{\theta}_i!} \times \theta_i \right) + (\theta_i - \theta_i^*) \right) \times \prod_{M_i \in \mathcal{M}^\bullet} \theta_i \quad (6.8)$$

Le nombre total $\mathcal{C}_{\mathcal{M}}$ de combinaisons existantes pour un système donné, exprimé par l'équation 6.8, correspond à un nombre maximal d'évaluations à opérer. Il est établi en

fonction du nombre k de machines implantées sur le système flow-shop contrôlé ainsi que de la répartition de la séquence opératoire de chaque machine en termes de durée, d'unités de temps de latence et d'unités de temps de traitement confiées au contrôleur central :

$$C_{\mathcal{M}} = f(k; \forall M_i \in \mathcal{M} : \theta_i, \theta_i^*, \overline{\theta_i}) \tag{6.9}$$

Pour un système donné, il existe potentiellement plusieurs combinaisons valides satisfaisant le critère d'optimalité. Ainsi, pour la gamme opératoire présentée par le tableau 6.1, nous proposons dans le tableau 6.3 deux solutions de synchronisation des machines qui permettent au système flow-shop contrôlé d'atteindre la borne τ_{max} sur le délai d'inter-sorties de pièces.

1	0	0	1	0	1	1	0	0	1	0	1
0	1	1	0	-	-	-	0	1	-	1	0
-	-	-	0	1	0	0	1	0	-	-	-
1	1	1	1	1	1	1	1	1	1	1	1

TAB. 6.3: Proposition de combinaisons optimales

Les résultats d'exécution correspondant aux combinaisons exprimées dans le tableau 6.3 sont présentés par la figure 6.4. Les combinaisons droite et gauche sont respectivement associées aux diagrammes de Gantt situés en parties haute et basse de la figure. Contrairement à la combinaison de droite, celle de gauche impose l'utilisation du mécanisme de préemption en suspendant durant une unité de temps le traitement du message de requête inscrit dans le cycle opératoire de la machine de production implantée en deuxième position sur le système.

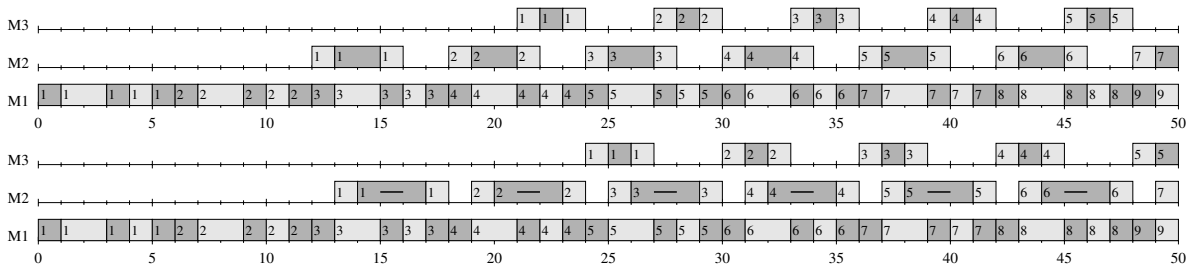


FIG. 6.4: Diagrammes d'exécution de machines synchronisées

Les diagrammes de Gantt confirment que le contrôleur central est en permanence en activité et que le système produit à la cadence maximale avec les deux combinaisons proposées. Les machines de production respectent toutes la période fondamentale τ_{max} . Contrairement au comportement des systèmes non-contrôlés, les différents démarrages du cycle opératoire des machines ne sont pas uniquement conditionnés par la présence d'une pièce disponible à leur entrée. Ils s'effectuent à des instants préalablement définis en fonction de la combinaison de séquences opératoires choisie et de la période fondamentale τ_{max} que chacune des ressources doit respecter.

Un processus de recherche simple et naturel à mettre en œuvre consiste alors à parcourir l'ensemble des combinaisons existantes de manière exhaustive et à stopper l'évaluation dès l'atteinte d'une combinaison satisfaisant le critère d'optimalité sans avoir à explorer la totalité des possibilités. Bien que le nombre \mathcal{C}_M puisse être très grand, plusieurs combinaisons optimales existent potentiellement. Mais il se peut aussi qu'aucune combinaison optimale ne puisse être trouvée. Le cas échéant, le processus effectue un parcours de toutes les possibilités pour ne retenir qu'une combinaison aboutissant au meilleur délai d'inter-sorties de pièces.

Ce mode d'exploration peut évidemment être coûteux en termes de temps et de ressources. Il faut néanmoins prendre en considération le fait que, dans un contexte déterministe des temps opératoires et du comportement général des machines, cette évaluation n'est à opérer qu'une fois au moment de la configuration du mode de consommation des messages de requête par le contrôleur central. Le coût de recherche d'une combinaison optimale ne représente donc pas un véritable problème puisqu'elle ne relève pas d'une adaptation dynamique de l'activité du contrôleur durant l'exécution du système flow-shop contrôlé.

Pour certaines gammes opératoires particulières, il est possible de détecter a priori et simplement l'impossibilité d'obtenir une combinaison optimale sans avoir recours à une exploration de l'ensemble des combinaisons. Les possibilités de trouver une combinaison optimale sont notamment limitées lorsqu'aucune des machines ne dispose d'unités de temps de latence dans sa séquence opératoire et que le contrôleur est aussi un goulet d'étranglement. Toutes les machines et le contrôleur du système sont ainsi critiques du point de vue du délai optimal τ_{max} d'inter-sorties de pièces. La synchronisation ne repose alors que sur les rotations successives effectuées à partir de la séquence initiale des machines et sur la correspondance en terme de durée de chaque opération locale

d'une machine donnée avec une opération distante d'une autre machine pour éviter les envois concurrents de messages de requête au contrôleur.

À titre d'exemple, nous pouvons citer le cas simple d'un système constitué de deux machines M_1 et M_2 ayant respectivement pour séquence initiale $\eta_1 = |1010|$ et $\eta_2 = |1100|$ dont le délai optimal τ_{max} d'inter-sorties de pièces est borné à quatre unités de temps. Nous considérons dans cet exemple que la phase de transfert entre les deux machines ne constitue pas un goulet d'étranglement :

$$\tau_{max} = 4 : \begin{cases} \sum_{M_i \in \mathcal{M}} \theta_i^* = \tau_{max} \\ \forall M_i \in \mathcal{M} : \theta_i = \tau_{max} \end{cases} \quad (6.10)$$

La décomposition 6.11 illustre les déclinaisons à partir de la séquence opératoire initiale η_i de chacune des machines. Quatre séquences opératoires \mathcal{C}_i peuvent être déclinaées pour chaque machine. Cependant, une duplication des séquences opératoires déclinaées pour la machine M_1 est constatée en raison de l'alternance du type d'opération à chaque unité de temps et de l'existence d'une sous-séquence $|10|$ répétée. Le nombre \mathcal{C}_1 peut ainsi être réduit à deux déclinaisons.

$$M_1 : \begin{cases} |1 & 0 & 1 & 0| \\ |1 & 0 & 1 & 0| \\ |1 & 0 & 1 & 0| \\ |1 & 0 & 1 & 0| \end{cases} \Rightarrow \begin{cases} |1 & 0 & 1 & 0| \\ |0 & 1 & 0 & 1| \\ |1 & 0 & 1 & 0| \\ |0 & 1 & 0 & 1| \end{cases} \quad (6.11)$$

$$M_2 : \begin{cases} |1 & 1 & 0 & 0| \\ |1 & 1 & 0 & 0| \\ |1 & 1 & 0 & 0| \\ |1 & 1 & 0 & 0| \end{cases} \Rightarrow \begin{cases} |1 & 1 & 0 & 0| \\ |0 & 1 & 1 & 0| \\ |0 & 0 & 1 & 1| \\ |1 & 0 & 0 & 1| \end{cases}$$

Les combinaisons possibles sont au nombre de huit, soit $\mathcal{C}_1 \times \mathcal{C}_2$. Aucune d'entre-elles ne représente une solution optimale. De manière évidente, il n'existe aucune opération locale affectée à la machine M_1 permettant l'exécution simultanée de la première opération affectée à la machine M_2 nécessitant un traitement par le contrôleur et dont la

durée est fixée à deux unités de temps. Le tableau 6.4 détaille l'ensemble des combinaisons possibles. Il met en évidence l'inexistence de combinaisons permettant d'éviter la concurrence d'accès au contrôleur central.

1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	0	0	0	1	1	0	0	0	1	1	1	0	0	1
2	1	1	0	1	1	2	0	1	0	2	1	2	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	1	0	0	0	1	1	0	0	0	1	1	1	0	0	1
1	2	0	1	0	2	1	1	0	1	1	2	1	1	0	2

TAB. 6.4: *Détail des combinaisons pour un cas sans solution optimale*

L'ensemble $\mathcal{C}_{\mathcal{M}}$ des combinaisons pour cette gamme opératoire aboutit à un délai τ d'inter-sorties de pièces étendu à cinq unités de temps, soit une dégradation de 25 % du degré de productivité du système.

6.2.4 Répartition de la fonction de coordination

La synchronisation des machines de production par la méthode proposée impose que la fonction de coordination ne soit pas uniquement assurée par le contrôleur central. En effet, la date de début de chacun des cycles opératoires successifs qu'exécute une machine doit être maîtrisée afin de respecter strictement la période fondamentale τ_{max} du système flow-shop contrôlé. Les machines sont alors chargées de gérer directement leur date d'entrée en production et leur durée d'inactivité à observer entre deux pièces successives.

Ce contrôle ne peut se faire qu'au niveau des machines en raison de l'inexistence d'interaction avec le contrôleur central lors du traitement d'une opération locale. Il peut toutefois être envisagé dans la mesure où la séquence opératoire de chaque machine serait toujours initiée par une opération distante confiée au contrôleur.

Sur un plan pratique, la répartition de la fonction de coordination entre les machines et le contrôleur implique l'implémentation dans les programmes de commande des machines de routines spécifiques dédiées à la planification de l'enchaînement des cycles opératoires.

6.2.5 Critères complémentaires pour la sélection d'une combinaison d'une gamme opératoire

Dans le cas où plusieurs combinaisons optimales existent pour un système donné, il peut être intéressant d'évaluer la pertinence que représente une combinaison optimale par rapport à une autre en prenant en compte des critères complémentaires à l'unique respect de la période fondamentale τ_{max} . Dans le cadre d'une implémentation sur un système de production réel, nous proposons de considérer deux critères secondaires dans le choix d'une combinaison optimale. La prise en compte de ces deux critères complémentaires est aussi applicable dans le cas où aucune combinaison optimale ne peut être trouvée mais que le meilleur délai d'inter-sorties de pièces soit atteignable avec plusieurs combinaisons.

Tout d'abord, il nous paraît pertinent de limiter les préemptions de messages de requête afin de minimiser les permutations de contexte influant sur le temps de réponse du contrôleur [Jiang and Cheng, 2001] [Zhou and Petrov, 2006]. Les permutations de contexte se soldent par une consommation de temps liée à la gestion interne d'un système implémentant le mécanisme de préemption [Snyder et al., 1995]. Le terme *overhead* est souvent utilisé pour désigner ce temps inactif du point de vue du système contrôlé. Il est donc avantageux de sélectionner parmi les combinaisons optimales celle qui minimise les interruptions dans le traitement des messages de requête. En reprenant les exemples du tableau 6.3, la combinaison de droite associée à la partie haute de la figure 6.4 représente une meilleure solution par rapport à celle de gauche.

Nous proposons également de favoriser la combinaison qui permet de terminer au plus tôt l'exécution des opérations pour chacune des machines du système en avançant au maximum les dates de début et de fin de traitement des messages de requête par le contrôleur central. Il s'agit d'introduire une forme de tolérance aux retards accumulés durant le déroulement du cycle opératoire d'une machine. Le tableau 6.5 montre, pour une combinaison donnée, un exemple de compensation d'un retard survenant au cours d'une opération exécutée localement. La seconde opération de la première machine, initialement fixée à une unité de temps, est prolongée d'une unité de temps sans pour autant pénaliser la cadence de production optimale du système. Le traitement du message de requête correspondant à la troisième opération de la dernière machine est alors repoussé.

$$\begin{array}{ccccc}
 1 & 0 & 1 & 0 & - \\
 0 & 1 & 0 & 0 & 0 \\
 - & 0 & 0 & 1 & - \\
 \hline
 1 & 1 & 1 & 1 & 0
 \end{array}
 \Rightarrow
 \begin{array}{ccccc}
 1 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 0 \\
 - & 0 & 0 & \rightarrow & 1 \\
 \hline
 1 & 1 & 0 & 1 & 1
 \end{array}$$

TAB. 6.5: Exemple d'une combinaison tolérante aux retards

Bien évidemment, le mécanisme de tolérance aux retards n'est applicable que dans certains cas de figure particuliers établis en fonction de la séquence opératoire affectée à chacune des machines de production. Il est en outre impératif que le contrôleur ne soit pas le goulet d'étranglement du système.

6.2.6 Principales conséquences de la synchronisation des machines de production

La synchronisation des machines de production à partir d'une combinaison valide aboutit, au-delà de la satisfaction de l'objectif premier d'optimisation du délai d'inter-sorties de pièces, à certaines conséquences sur le déroulement d'un système contrôlé.

Le premier constat porte sur la rapidité d'établissement du régime stationnaire. En effet, la date d'exécution des différentes unités de temps de la séquence opératoire affectée à une machine est figée pour toutes les pièces. Chaque machine du système reproduit de manière identique l'exécution de son cycle opératoire dès la prise en charge de la première pièce. En comparaison des résultats d'exécution relevés lors de la phase de simulation, le système effectue son passage en régime stationnaire dès le début de son entrée en production.

Une seconde conséquence se situe au niveau des encours de pièces entre deux machines successives. Le volume de pièces en attente à l'entrée d'une machine est très réduit et n'excède pas une pièce. À chaque instant de l'exécution du système, le nombre total de pièces en cours de production est ainsi limité. Dans le contexte d'un système réel, cette conséquence peut revêtir une certaine importance selon la nature des produits fabriqués en limitant, par exemple, le volume de composants critiques disposés sur le procédé à chaque instant. Elle est également intéressante dans le cas de dysfonctionnement du système engendrant la mise au rebut des pièces présentes sur le procédé au moment où l'anomalie est détectée.

6.3 Conclusion

Les conditions d'atteinte du délai optimal τ_{max} d'inter-fins de travaux, et donc d'inter-sorties de pièces, pour un système flow-shop contrôlé ont été abordées dans ce chapitre. Nous proposons une méthode permettant d'aboutir à la maximisation de la cadence de production. Elle ne garantit toutefois pas de manière systématique l'optimalité du délai d'inter-sorties de pièces qui n'est, en effet, pas toujours atteignable.

La méthode proposée est basée sur la synchronisation de la séquence opératoire et sur la date d'entrée en activité des différentes machines de production implantés sur un système. Elle exploite également les unités de temps de latence inscrites dans la séquence opératoire de chaque machine pour réduire, voire annuler, les accès concurrents au contrôleur central. La date de traitement par le contrôleur de chaque unité de temps d'un message de requête est ainsi prédéfinie. L'absence d'unités de temps de latence restreint alors les possibilités de trouver une solution optimale.

Les choix quant à la synchronisation des machines sont effectués avant le début d'exécution du système contrôlé. Notre méthode correspond ainsi à un ordonnancement hors ligne. Ils considèrent un comportement déterministe du système. Cependant, des retards détectés au cours du déroulement du cycle opératoire peuvent être compensés sous certaines conditions.

Une contrainte essentielle induite par la méthode de synchronisation des machines se situe dans la répartition de la fonction de coordination qui ne peut plus être uniquement assurée par le contrôleur central. La méthode proposée ne constitue donc pas une véritable solution à la problématique posée par les systèmes que nous considérons où seul le contrôleur central supporte la fonction de coordination. Elle permet cependant d'évaluer la faisabilité d'un ordonnancement optimal et de déterminer a priori la cadence maximale que le système flow-shop contrôlé peut atteindre. De plus, le principe sur lequel repose la méthode jette les principaux fondements d'un ordonnancement efficace exécutable en ligne.

Chapitre 7

Propositions pour un ordonnancement en ligne des messages de requête

Sommaire

7.1	Intérêt d'un ordonnancement en ligne	172
7.2	Choix des règles de tri utilisées	173
7.3	Particularité du contexte d'utilisation des règles de tri .	176
7.3.1	Calcul des priorités	177
7.3.2	Sélection de la machine la plus prioritaire	179
7.4	Validation des règles de tri par simulation	179
7.4.1	Cas d'une configuration bornée par une machine de pro- duction	180
7.4.2	Cas d'une configuration bornée par le contrôleur central .	183
7.4.3	Cas d'une configuration sans possibilité d'atteinte de l'op- timalité	186
7.5	Conclusion	188

Dans le chapitre précédent, nous avons présenté une méthode d'ordonnancement destinée à synchroniser l'activité des machines de production fixant ainsi la date de traitement par le contrôleur pour chaque unité de temps d'un message de requête avant le démarrage du système. Les accès concurrents au contrôleur sont alors maîtrisés. La méthode proposée est basée sur une évaluation a priori du délai minimum d'inter-sorties de pièces que le système est en mesure d'atteindre. Cette méthode se résume à un ordonnancement hors ligne de l'activité du contrôleur central à partir d'une table décrivant statiquement l'ordre des messages de requête à traiter et donc des machines à servir.

Dans ce chapitre, nous proposons de résoudre la problématique par la mise en œuvre d'un ordonnancement en ligne utilisant une gestion dynamique des priorités. Il s'agit alors de sélectionner à chaque unité de temps le message de requête le plus urgent en prenant en compte le mécanisme de préemption. Notre proposition est basée sur l'utilisation de règles de tri couramment utilisées dans les systèmes informatiques temps réels, domaine dans lequel nous pouvons, dans une certaine mesure, situer nos travaux en raison des contraintes de temps qui s'appliquent aux procédés étudiés. En effet, les applications temps réel sont celles où le facteur de temps est la principale contrainte à respecter et où ce facteur est prépondérant pour évaluer la qualité du service [Cottet et al., 2000].

Alors que la méthode par synchronisation des machines de production impose une répartition de la fonction de coordination, notre objectif est ici de répondre exactement à la problématique posée en respectant la structure de contrôle initialement considérée concentrant la fonction de coordination sur le seul contrôleur central. La période fondamentale τ_{max} reste néanmoins évaluable à partir de la méthode par synchronisation des machines de production.

7.1 Intérêt d'un ordonnancement en ligne

À la différence d'un ordonnancement hors ligne, un ordonnancement en ligne évalue la priorité des tâches à traiter durant l'exécution du système. Parmi les ordonnancements en ligne, deux types principaux peuvent être déclinés. Lorsque la priorité d'une tâche évolue au cours de la durée de son activation, l'ordonnancement est dit à *priorités dynamiques* par opposition à un ordonnancement à *priorités statiques* où la priorité des tâches reste figée [Stankovic et al., 1985].

Dans un système informatique temps réel dont le fonctionnement est assujéti à l'évolution dynamique du procédé à contrôler, l'ordonnancement des tâches chargées de la surveillance et de la commande de ce procédé joue un rôle capital [Cottet et al., 1999]. L'utilisation d'un algorithme d'ordonnancement hors ligne aboutit généralement à de meilleures performances lorsque le comportement de l'environnement est statique et déterministe [Kizilişik, 1999]. Cependant, l'application d'un ordonnancement en ligne permet de réagir dynamiquement au comportement du système contrôlé. En comparaison d'un ordonnancement hors ligne, reposant sur une approche prédictive, il apporte une réponse plus efficace aux événements survenant au cours de l'exécution du procédé [Sabuncuoglu and Karabük, 1999].

Un système de production industriel est en parfois soumis à des dysfonctionnements de natures diverses entraînant une variation de la durée de déroulement du cycle opératoire des machines. La considération de cet indéterminisme rend difficile l'utilisation d'un ordonnancement hors ligne. Un ordonnancement en ligne offre, quant à lui, une plus forte réactivité en permettant une compensation totale ou partielle de la déviation du comportement d'un système de production. Il répond mieux au problème de pilotage en temps réel de machines de production.

7.2 Choix des règles de tri utilisées

Nos propositions d'ordonnancement en ligne des messages de requête sont basées sur les règles de tri *EDD*, ou règle de Jackson, et *LST*. Dans le contexte de leur adaptation au domaine des systèmes temps réels, les règles de tri *EDD* et *LST* sont le plus souvent désignées par *EDF* [Liu and Layland, 1973] [Chetto and Chetto, 1989] [Chetto, 1990], pour *earliest deadline first*, et *LLF* [Sorenson, 1974] [Dhall, 1977], pour *least laxity first*.

Les algorithmes *EDF* et *LLF* sont les plus couramment utilisés dans le cadre d'un ordonnancement en ligne à priorités dynamiques pour un système temps réel mettant en jeu des tâches périodiques exécutées par une unique ressource généralement désignée par le terme *processeur*. Dans la suite de ce chapitre, nous considérons leur mise en œuvre dans ce contexte en raison du comportement périodique des machines de production et de l'utilisation du mécanisme de préemption pour ordonnancer l'activité du contrôleur central.

L'algorithme *EDF* est basé sur l'évaluation de la date d'échéance des tâches actives.

La tâche de plus haute priorité est celle dont l'échéance est la plus proche. L'optimalité de l'algorithme *EDF* a été démontrée pour une application temps réel composée de tâches indépendantes à échéances inférieures ou égales aux périodes [Dertouzos, 1974] [Labetoulle, 1974]. Il a cependant été montré que l'algorithme *EDF* adopte un comportement instable en cas de surcharge du processeur [Delacroix, 1994].

Lorsque l'échéance des tâches coïncide avec l'activation suivante, elles sont dites à *échéance sur requête*. Dans une telle configuration de tâches, une condition nécessaire et suffisante de faisabilité de l'ordonnancement est donnée par le taux U d'utilisation du processeur. Il est donné par le rapport de la durée d'exécution C_i dans le pire cas et de la période P_i de chaque tâche T_i mise en jeu dans le système. L'algorithme *EDF* permet une utilisation maximale du processeur :

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \leq 1 \quad (7.1)$$

Pour des tâches quelconques, le résultat de l'équation 7.1 se limite à une condition nécessaire. La condition suffisante se détermine à partir de l'échéance D_i , avec $D_i \leq P_i$, fixée pour chacune des tâches :

$$U = \sum_{i=1}^n \frac{C_i}{D_i} \leq 1 \quad (7.2)$$

À titre d'exemple, nous considérons un jeu de trois tâches à échéance sur requête dont les caractéristiques sont indiquées dans le tableau 7.1. Le taux U d'utilisation du processeur s'évalue à 100 % vérifiant ainsi la condition nécessaire et suffisante donnée par l'équation 7.1.

Tâche T_i	Capacité C_i	Période P_i
T_1	4	16
T_2	2	8
T_3	3	6

TAB. 7.1: Jeu de tâches évalué

Nous présentons sur la figure 7.1 le résultat d'exécution obtenu avec l'algorithme *EDF* en mode préemptif. Les différentes activations de chaque tâche sont marquées par une flèche verticale. Seule la tâche T_1 fait l'objet d'une préemption. Par conformité avec les

diagrammes de Gantt présentés dans le document, les unités de temps correspondant à une préemption sont représentées par un trait horizontal.

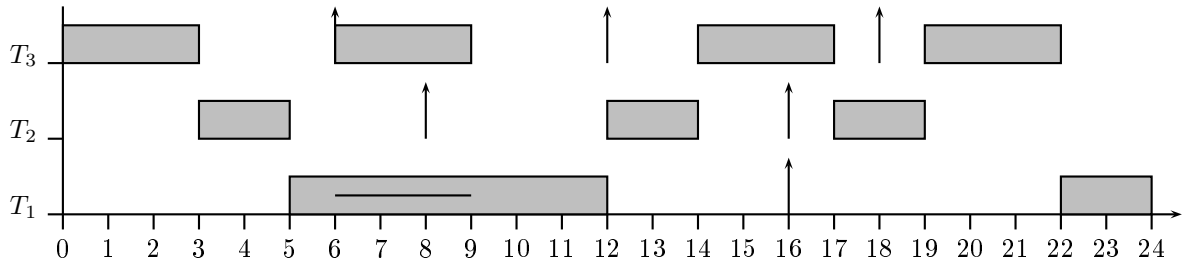


FIG. 7.1: Exemple d'exécution avec l'algorithme EDF

L'algorithme *LLF* repose sur une affectation de priorité aux tâches en fonction de la marge dont elles disposent avant l'échéance d'exécution. Cette marge, désignée par le terme *laxité dynamique*, correspond à la différence entre le délai avant échéance et la durée d'exécution restante. L'algorithme *LLF* présente les mêmes conditions de faisabilité d'ordonnancement que celles données pour l'algorithme *EDF* par les équations 7.1 et 7.2. Tout comme l'algorithme *EDF*, son optimalité pour des applications temps réel composées de tâches indépendantes à échéances inférieures ou égales aux périodes a été démontrée [Mok, 1983].

La figure 7.2 montre le résultat d'exécution du jeu de tâches donné dans le tableau 7.1 obtenu avec l'algorithme *LLF* en mode préemptif.

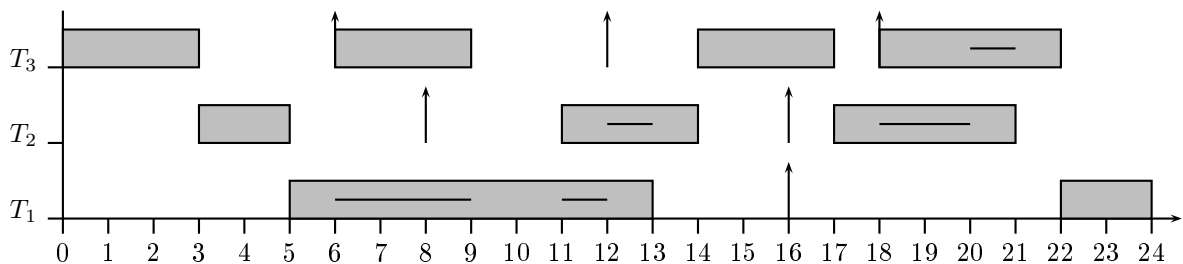


FIG. 7.2: Exemple d'exécution avec l'algorithme LLF

La comparaison des figures 7.1 et 7.2 montre qu'en mode mono-processeur l'algorithme *LLF* est moins efficace que l'algorithme *EDF* dans la mesure où il engendre un plus grand nombre de changements de la tâche en cours d'exécution par le processeur. Ce

passage d'une tâche à une autre est appelé une *permutation de contexte*. Des variantes de l'algorithme *LLF* ont cependant été proposées afin de le rendre plus performant en réduisant le nombre de permutations de contexte [Oh and Yang, 1998].

Bien que nous ayons limité la présentation des deux diagrammes de Gantt sur les vingt-quatre premières unités de temps, la période d'étude pour un ordonnancement pour une application temps réel s'évalue en fonction du *PPCM* des différentes périodes du jeu de tâches, soit $[0, PPCM(P_i)]$, lorsque la première activation de chacune des tâches s'effectue à l'initialisation du système [Leung and Merrill, 1980]. Pour le jeu de tâches évalué et présenté dans le tableau 7.1, la période d'étude correspond à un intervalle de quarante-huit unités de temps.

L'application des algorithmes *EDF* et *LLF* au temps réel peut être ramenée à un problème d'optimisation où la fonction objectif vise la minimisation du retard algébrique maximal L_{max} [Buttazzo, 1997]. Dans un contexte mono-processeur, l'expression du problème devient alors $1|r_j, pmtn|L_{max}$ selon la notation de Graham. Bien que le problème principal que nous cherchons à résoudre dans le cadre des travaux présentés s'écrive $F, C1/pmtn_c, permu, block, r_j/C_{max}$ et concerne la minimisation du makespan C_{max} , nous répondons à la problématique posée par la mise en œuvre d'un ordonnancement adapté à l'activité du contrôleur en visant la minimisation du retard algébrique L_{max} . Le choix des algorithmes *EDF* et *LLF* comme base de réflexion repose sur cette argumentation.

7.3 Particularité du contexte d'utilisation des règles de tri

Dans un système temps réel, la règle de tri mise en œuvre s'appuie sur des caractéristiques statiques et dynamiques propres à chacune des tâches ainsi que sur l'état courant du système pour calculer la priorité à affecter à chaque tâche active à un instant donné et ainsi sélectionner celle qui doit être exécutée en désignant la plus prioritaire. La particularité du contexte d'utilisation que nous proposons consiste à évaluer l'urgence d'un message de requête non pas en fonction de ses caractéristiques propres mais à partir de la situation opératoire de la machine émettrice. En effet, nous avons montré que toutes les machines de production sont soumises à la fréquence fondamentale τ_{max} et que son

respect garantit l'atteinte du délai optimal d'inter-sorties de pièces du système flow-shop contrôlé.

Par analogie aux systèmes temps réels, nous considérons un message de requête soumis au contrôleur central comme une tâche du système. Chaque envoi d'un message de requête est alors une nouvelle instantiation de la tâche correspondante. De plus, la date de première activation d'un message de requête donné est dépendante de la date d'entrée en production de la machine émettrice.

7.3.1 Calcul des priorités

Alors que pour les algorithmes *EDF* et *LLF* la priorité d'une tâche est dynamiquement évaluée à chaque instant de l'exécution du système en fonction de son état, l'urgence d'un message de requête doit être considérée dans un contexte global d'exécution du cycle opératoire de la machine de production émettrice. Nos propositions pour un ordonnancement en ligne de l'activité du contrôleur reposent sur la prise en compte de l'échéance et de la laxité appliquée au cycle opératoire de la machine concernée et non pas directement au traitement du message de requête par le contrôleur. Nous introduisons donc les notions d'*échéance opératoire* et de *laxité opératoire* pour exprimer, pour une machine donnée, la date de fin d'exécution du cycle opératoire en cours et la marge disponible avant cette échéance.

L'échéance opératoire pour une machine M_i émettrice d'un message de requête est évaluée à partir de trois paramètres :

- la date β_0^i d'entrée en production,
- la pièce J_j en cours de traitement,
- la période fondamentale τ_{max} du système flow-shop contrôlé.

L'échéance opératoire d'une machine de production M_i pour une pièce J_j est notée φ_j^i . Par extension de la notation, l'échéance opératoire pour la pièce en cours de traitement à l'instant t s'écrit $\varphi_j^i(t)$. Le calcul des dates correspondant aux échéances opératoires peut être effectué pour l'ensemble des pièces à produire dès l'entrée en production de la machine :

$$\forall J_j \in \mathcal{J}, \forall M_i \in \mathcal{M} : \varphi_j^i = \beta_0^i + (\tau_{max} \times j) \quad (7.3)$$

La figure 7.3 montre le principe d'application du calcul de l'échéance opératoire à une machine de production. Chaque tâche du diagramme correspond à l'ensemble de

la séquence opératoire de la machine. Dans l'exemple présenté, la machine respecte ses échéances et ne pénalise donc pas le délai optimal τ_{max} d'inter-sorties de pièces du système.

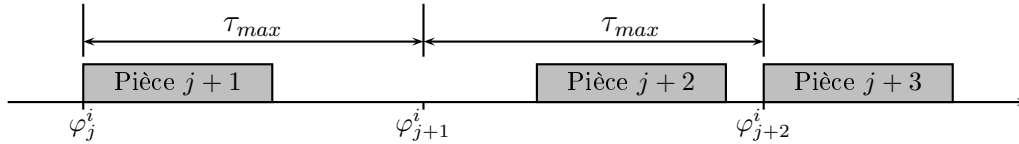


FIG. 7.3: Évaluation des échéances opératoires pour une machine

La laxité opératoire dont dispose une machine M_i à l'instant t nécessite de connaître la durée cumulée des opérations restant à exécuter pour achever le traitement d'une pièce J_j à cet instant. Nous désignons cette durée par le terme *temps opératoire résiduel* et la notons $\gamma_j^i(t)$. La laxité opératoire d'une machine de production est notée $\phi_j^i(t)$ et son calcul est évidemment basée sur la proximité de l'échéance opératoire déterminée pour la pièce en cours de traitement :

$$\forall J_j \in \mathcal{J}, \forall M_i \in \mathcal{M} : \phi_j^i(t) = \varphi_j^i - t - \gamma_j^i(t) \quad (7.4)$$

Les algorithmes *EDF* et *LLF* ont souvent servi de base de travail pour l'implémentation de règles de tri adaptées à la résolution de problèmes particuliers ou, plus simplement, pour en améliorer les performances [Hildebrandt et al., 1999] [Cho et al., 2002] [Uthaisombut, 2004] [Anderson et al., 2005]. Les règles de tri que nous proposons s'appuient sur une prise en compte d'une échéance et d'une laxité dynamique opératoires, que nous introduisons, pour établir le degré d'urgence des messages de requête. Ces caractéristiques ne sont pas directement liées aux tâches à exécuter mais conditionnent néanmoins l'activité du contrôleur. Les messages de requête ne sont en effet pas directement affectés d'une échéance ou d'une laxité propres et leur durée de traitement par le contrôleur central n'est pas prise en compte dans l'évaluation de leur degré d'urgence. Les règles de tri proposées, que nous nommons *EoDF* et *LoLF* pour *earliest operating deadline first* et *least operating laxity first*, relèvent donc d'une adaptation originale des algorithmes *EDF* et *LLF*. Le test de faisabilité basé sur les équations 7.1 et 7.2 n'est donc pas applicable dans le contexte de nos travaux.

7.3.2 Sélection de la machine la plus prioritaire

Que la règle de tri utilisée soit basée sur l'algorithme *EDF* ou *LLF*, l'urgence d'un message de requête en cours ou en attente de traitement par le contrôleur à un instant t s'évalue en fonction de la situation globale de la machine émettrice. L'évaluation s'effectue par rapport à l'échéance opératoire de fin de traitement de la pièce associée. La priorité est ainsi plus directement affectée à la machine qu'au message de requête qu'elle a soumis au contrôleur. À chaque instant t de l'exécution du système, le message en provenance de la machine la plus prioritaire, désignée par $M_\diamond(t)$, devient le plus urgent. Il est alors traité par le contrôleur :

$$\text{Pour la règle de tri } EoDF : M_\diamond(t) = \min_{M_i \in \mathcal{M}} \{\varphi_j^i(t)\} \quad (7.5)$$

$$\text{Pour la règle de tri } LoLF : M_\diamond(t) = \min_{M_i \in \mathcal{M}} \{\phi_j^i(t)\}$$

Le mécanisme d'identification de la machine la plus prioritaire à l'instant t , et donc du message de requête à exécuter, est d'une grande simplicité. Sa complexité dépend uniquement du nombre de machines de production implantées sur le système considéré. Elle est ainsi en $O(n)$.

7.4 Validation des règles de tri par simulation

Les règles de tri que nous proposons reposent sur les algorithmes *EDF* et *LLF* dont les performances sont connues et démontrées. Un travail de simulation reste néanmoins nécessaire afin de vérifier l'efficacité des règles de tri *EoDF* et *LoLF*. Nous présentons dans cette section, sous forme de diagrammes de Gantt, des résultats de simulation permettant de visualiser l'enchaînement des opérations sur les différentes machines de production constatés avec chacune des deux règles de tri. L'objectif est de confronter leur efficacité mais aussi de comparer le niveau de performance obtenu pour les systèmes simulés avec celui atteint avec les règles de tri courantes utilisées lors de la phase de simulation décrite dans le cinquième chapitre.

Les résultats de simulation sont d'abord exprimés pour un système dont le délai optimal τ_{max} d'inter-sorties de pièces est borné par une machine de production. La

seconde partie des résultats de simulation concerne le cas d'un système dont la cadence maximale est limitée par la charge du contrôleur central. Pour une confrontation aisée des résultats, les deux cas sont évalués à partir de gammes opératoires très similaires.

7.4.1 Cas d'une configuration bornée par une machine de production

Afin de valider l'efficacité d'un ordonnancement en ligne de l'activité du contrôleur et des règles de tri que nous proposons dans le cas où la cadence de production est bornée par une machine, nous considérons une gamme opératoire déclinées en douze opérations pour un ensemble de quatre machines de production. Sa décomposition, présentée dans le tableau 7.2, montre que la seconde machine constitue la ressource critique du système en limitant le délai optimal τ_{max} d'inter-sorties de pièces à onze unités de temps.

Opérations		M_i	θ_i	$\bar{\theta}_i$	τ_{max}
$\theta_{1,1}$	2	1	7	4	11
$\theta_{1,2}^*$	3				
$\theta_{1,3}$	2				
$\theta_{2,1}$	4	2	11	0	
$\theta_{2,2}^*$	4				
$\theta_{2,3}$	3				
$\theta_{3,1}$	2	3	5	6	
$\theta_{3,2}^*$	1				
$\theta_{3,3}$	2				
$\theta_{4,1}$	2	4	6	5	
$\theta_{4,2}^*$	2				
$\theta_{4,3}$	2				

TAB. 7.2: Gamme pour l'évaluation des règles de tri avec une borne mécanique

La gamme opératoire retenue est particulièrement intéressante car elle nous a servi à étudier, au cinquième chapitre, le cas d'un système dont la cadence est bornée par une machine de production et l'activité du contrôleur ordonnancée à l'aide des règles de tri *FIFO* et *LPT* non-préemptive. Aucune des deux règles de tri ne permet d'atteindre le délai optimal τ_{max} d'inter-sorties de travaux. Les résultats obtenus avec les règles de tri *EoDF* et *LoLF* sont présentés par les figures 7.4 et 7.5.

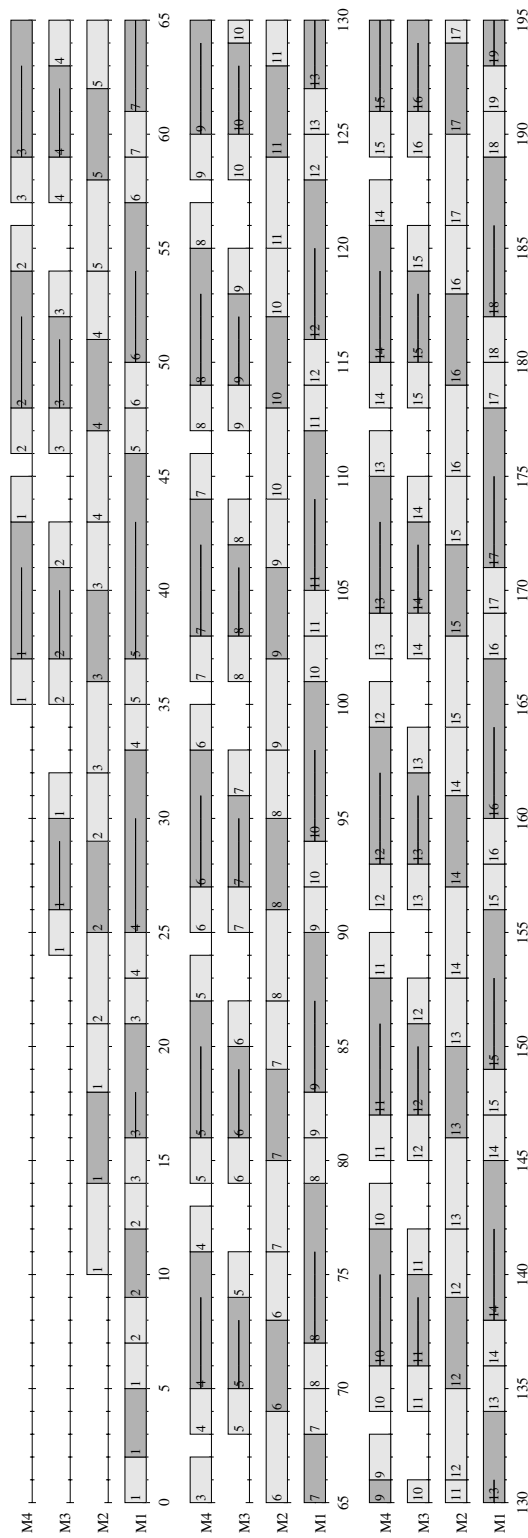


FIG. 7.4: Résultat avec une borne mécanique et la règle de tri EoDF

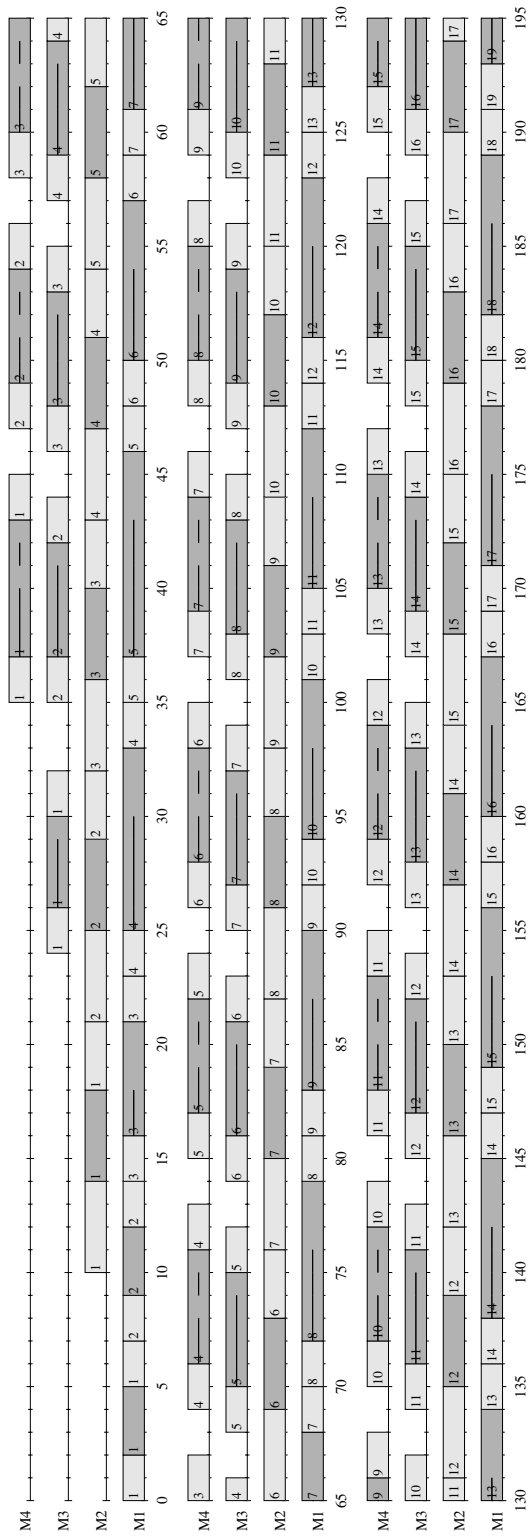


FIG. 7.5: Résultat avec une borne mécanique et la règle de tri LoLF

Dans les deux cas évalués, les diagrammes de Gantt montrent que le délai optimal τ_{max} d'inter-sorties de pièces est atteint. Les première et seconde machines, les plus lentes du système, détiennent en permanence une pièce. De plus, il est intéressant de constater que le nombre de pièces en encours n'excède jamais une unité et que le système simulé atteint très rapidement le régime stationnaire. Enfin, conformément au comportement caractéristique de l'algorithme *LLF*, la règle de tri qui en est dérivée engendre un plus grand nombre de permutations de contexte que celle basée sur l'algorithme *EDF*.

7.4.2 Cas d'une configuration bornée par le contrôleur central

Pour l'évaluation de l'efficacité des règles de tri proposées dans le cas d'un système dont le délai optimal τ_{max} d'inter-sorties de pièces est fixé par le contrôleur, nous reprenons la gamme opératoire définie dans le cinquième chapitre dans le cadre de simulations étendues. Seules les règles de tri *FIFO* et *Pull* non-préemptive ont alors permis d'aboutir à des résultats optimaux quant à la minimisation du makespan. Le détail de la gamme opératoire est rappelé dans le tableau 7.3.

Opérations		M_i	θ_i	$\bar{\theta}_i$	τ_{max}
$\theta_{1,1}$	2	1	7	3	10
$\theta_{1,2}^*$	3				
$\theta_{1,3}$	2				
$\theta_{2,1}$	2	2	8	2	
$\theta_{2,2}^*$	4				
$\theta_{2,3}$	2				
$\theta_{3,1}$	2	3	5	5	
$\theta_{3,2}^*$	1				
$\theta_{3,3}$	2				
$\theta_{4,1}$	2	4	6	4	
$\theta_{4,2}^*$	2				
$\theta_{4,3}$	2				

TAB. 7.3: Gamme pour l'évaluation des règles de tri avec une borne de contrôle

Les diagrammes de Gantt résultant de la simulation du système sont présentés dans les figures 7.6 et 7.7. Ils montrent que la cadence optimale est atteinte avec les deux règles de tri que nous proposons.

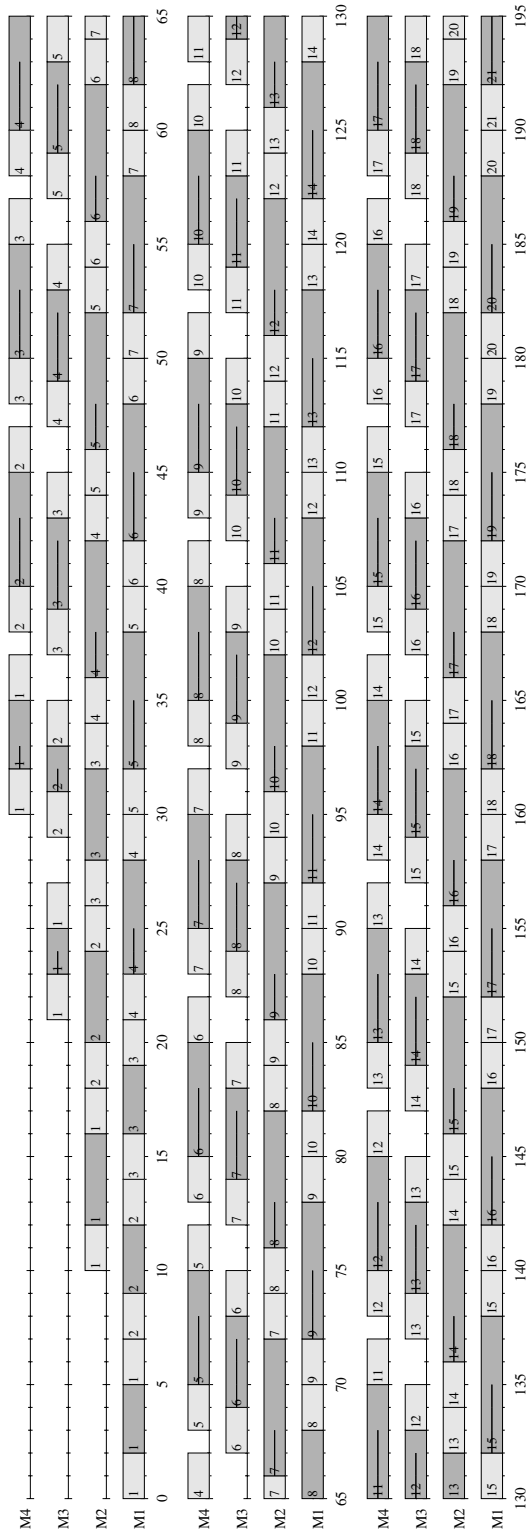


FIG. 7.6: Résultat avec une borne de contrôle et la règle de tri EoDF

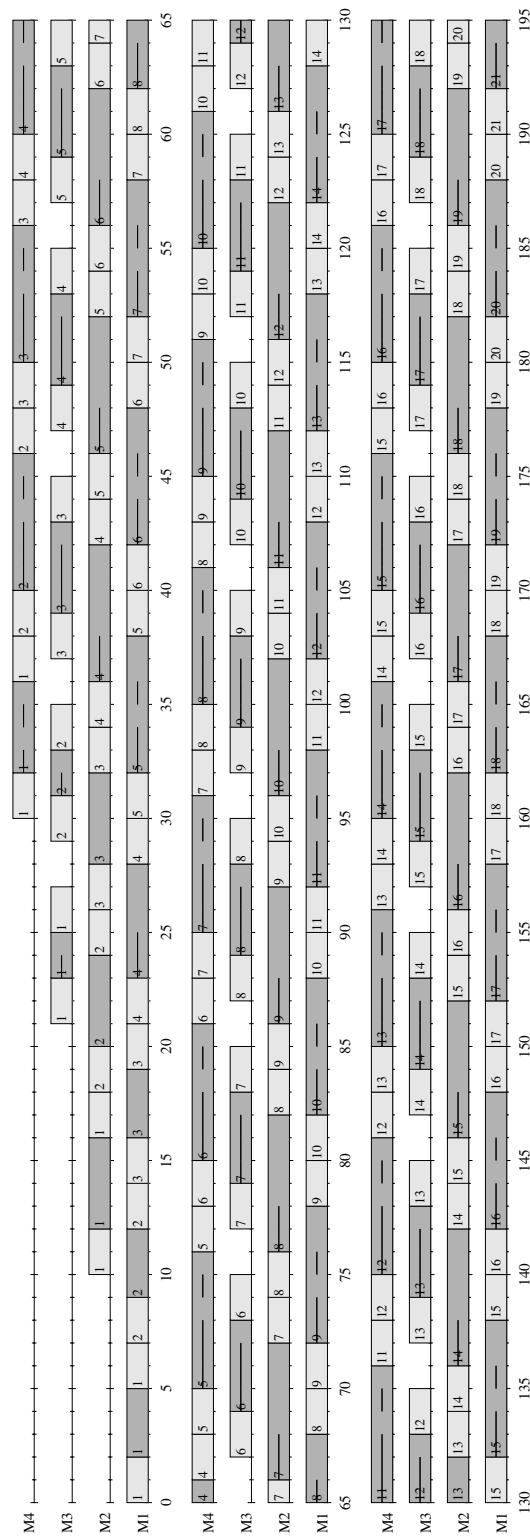


FIG. 7.7: Résultat avec une borne de contrôle et la règle de tri LoLF

Le niveau de performance globale atteint par le système est similaire à celui obtenu dans le cas de la configuration bornée par une machine de production. Les encours de pièces sont effet limités et le régime stationnaire est rapidement établi. Le nombre de permutations de contexte est également plus important lorsque la règle de tri *LoLF* est utilisée. Nous relevons également que les trois machines les plus lentes du système sont occupées par une pièce en permanence avec la règle de tri basée sur la laxité opératoire alors que seules les première et seconde sont dans cette situation avec la règle de tri basée sur l'échéance.

7.4.3 Cas d'une configuration sans possibilité d'atteinte de l'optimalité

Pour compléter la vérification par simulation des règles de tri que nous proposons, elles doivent être exploitées dans le cadre d'une gamme opératoire ne permettant pas d'atteindre la cadence optimale. Il s'agit alors de mesurer l'écart obtenu par rapport à la situation optimale et la capacité des règles de tri à permettre un cadencement du système au meilleur niveau atteignable en minimisant la dégradation de productivité.

Nous reprenons le cas particulier abordé au sixième chapitre concernant un système flow-shop constitué de deux machines se répartissant une gamme opératoire étalée sur huit unités de temps et composée de six opérations. Le délai optimal τ_{max} d'inter-sorties de pièces est conjointement fixé à quatre unités de temps par les deux machines et le contrôleur. Le tableau 7.4 présente le détail de la gamme opératoire.

Opérations		M_i	θ_i	$\bar{\theta}_i$	τ_{max}
$\theta_{1,1}$	1	1	4	0	4
$\theta_{1,2}^*$	1				
$\theta_{1,3}$	1				
$\theta_{1,4}^*$	1				
$\theta_{2,1}$	2	2	4	0	
$\theta_{2,2}^*$	2				

TAB. 7.4: Gamme pour l'évaluation d'une configuration sans atteinte de l'optimalité

Les résultats de simulation sont identiques pour les deux règles de tri proposées. Ils sont présentés par la figure 7.8.

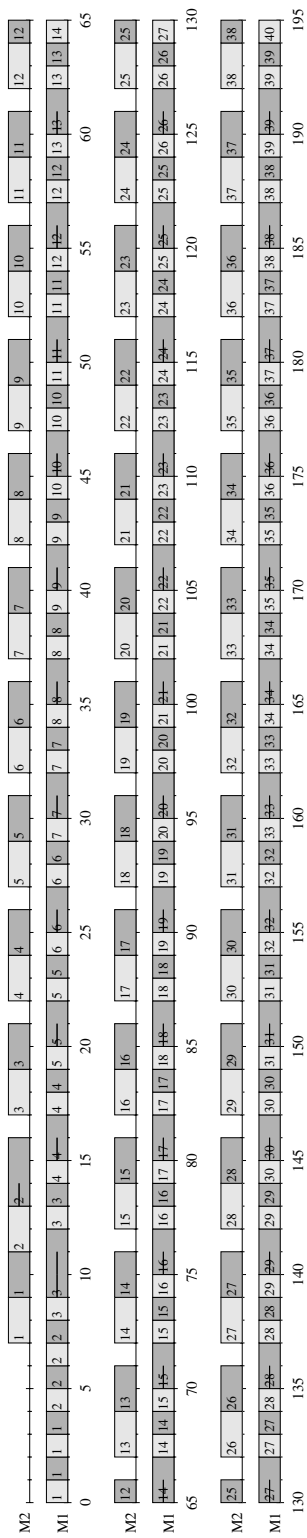


FIG. 7.8: Résultat du cas non-optimal pour les deux règles de tri

Pour le contexte particulier considéré, nous avons démontré au chapitre précédent que le délai optimal τ_{max} d'inter-sorties de pièces ne peut être atteint et que la dégradation minimale de la cadence est d'une unité de temps. Le diagramme de Gantt montre que ce meilleur cas est obtenu avec les deux règles de tri que nous proposons.

7.5 Conclusion

La mise en œuvre d'un ordonnancement en ligne pour organiser l'activité du contrôleur central apporte une certaine souplesse dans le pilotage des systèmes flow-shop considérés dans nos travaux. Un ordonnancement en ligne présente l'avantage sur les méthodes hors ligne de réagir dynamiquement aux événements, tels que l'arrivée ou le relâchement d'une pièce, survenant au cours du déroulement des cycles opératoires successifs des machines de production.

Nous présentons deux règles de tri, nommées *EoDF* et *LoLF*, permettant une sélection en ligne des messages de requête en fonction de l'état courant du système contrôlé tout en répondant à la contrainte d'une coordination centralisée posée par la structure de pilotage. Les règles de tri sont basées sur les algorithmes *EDF* et *LLF* largement répandus dans les systèmes temps réel. Notre choix est motivé par le fait que ces algorithmes sont connus pour être optimaux dans le cas du problème d'ordonnancement dynamique et préemptif relatif à un système composé d'une machine unique, ou mono-processeur, lorsqu'il s'agit de minimiser le retard algébrique maximal. Du point de vue de l'activité du contrôleur central, ce problème d'ordonnancement correspond à celui que nous cherchons à résoudre. Selon la notation de Graham, il s'exprime par $1|r_j, pmtn|L_{max}$.

L'application particulière des algorithmes *EDF* et *LLF* que nous proposons conditionne l'exécution des systèmes flow-shop contrôlés par le respect d'échéance ou de laxité opératoires imposées aux machines. Elle tend ainsi à contraindre les machines à produire en respectant la période fondamentale τ_{max} du système.

Les résultats obtenus par simulation laissent apparaître que les systèmes produisent à la meilleure cadence accessible et que le régime stationnaire est rapidement atteint. Ils permettent en outre de constater que le nombre de pièces en encours est limité et n'excède pas une unité pour les configurations évaluées.

Chapitre 8

Conclusion générale

Sommaire

8.1	Bilan des travaux	190
8.2	Travaux prospectifs	193

Les travaux présentés dans cette thèse trouvent leurs origines au cœur de l'industrie manufacturière dont l'activité consiste à assurer une production de masse. Les systèmes de production mis en œuvre relèvent alors souvent d'une organisation en ligne des machines pour répondre efficacement à des impératifs de productivité. Au cours des dernières décennies, ces systèmes d'assemblage de type flow-shop ont fait l'objet d'une automatisation de plus en plus importante et les moyens matériels et logiciels aujourd'hui exploités pour en assurer le pilotage représentent une part non négligeable des coûts de conception et de réalisation de ces procédés.

Les contraintes de temps auxquelles sont soumis les équipements informatiques en charge du contrôle et de la commande des machines ne peuvent pas être négligés lors de la phase d'étude de tels systèmes de production. Leur maîtrise est devenue un critère majeur de la performance globale des procédés pilotés. Un système de pilotage doit en effet toujours utiliser de manière optimale le matériel dont il assure le pilotage [Schild and Bussmann, 2007].

La particularité des systèmes flow-shop que nous considérons se situe dans l'architecture de pilotage déclinée sur deux niveaux. La coordination des machines est une fonction essentielle centralisée et assurée par un unique contrôleur dont l'activité consiste à répondre aux sollicitations des machines de production transmises via leur unité de commande. Les envois de requête fréquents au contrôleur au cours des cycles opératoires successifs des machines imposent l'obtention d'une réponse bornée dans le temps pour ne pas pénaliser l'écoulement du flux de matière à travers le système de production piloté.

L'objectif alors visé par les travaux exposés dans cette thèse ont principalement consisté à minimiser le délai τ d'inter-sorties de pièces assemblées en organisant au mieux l'activité du contrôleur central afin de permettre à des systèmes flow-shop contrôlés de produire à la meilleure cadence atteignable.

8.1 Bilan des travaux

Une organisation hiérarchisée du système de pilotage telle que décrite pose une problématique d'ordonnancement de l'activité du contrôleur central. Le problème globalement considéré du point de vue du système de production peut être exprimé sous la forme $F, C1/pmtn_c, permu, block, r_j/C_{max}$ selon la notation de Graham que nous étén-

dons pour la prise en compte d'une ressource de contrôle et de la restriction d'application du mécanisme de préemption au seul contrôleur. Cependant, en limitant le contexte du problème d'ordonnancement à l'activité du contrôleur central, nous pouvons réduire son expression à $1|r_j, pmtn|L_{max}$. En effet, bien que le problème d'ordonnancement du système global vise la minimisation du makespan C_{max} , celui du contrôleur isolé se résume à la satisfaction des échéances ou à la minimisation des retards dans la réponse apportée aux messages de requête.

À partir de relevés effectués sur des lignes d'assemblage réelles, un travail de modélisation a été réalisé à l'aide du formalisme des réseaux de Petri colorés temporisés. Il a permis d'aboutir à un modèle hautement paramétrable permettant son utilisation pour l'étude d'une large variété de systèmes de production en ligne. L'extension colorée des réseaux de Petri permet au modèle de conserver une taille restreinte et indépendante du nombre de machines déclarées et de la nature des séquences opératoires. La temporisation autorise le paramétrage des temps opératoires et de transfert de pièces entre deux machines successives. Le modèle produit constitue un outil dont l'utilisation est intéressante dans le cadre de l'analyse d'un système en cours de conception. Il peut également être une aide pour l'évaluation a priori des effets de la modification ou de la reconfiguration d'un procédé existant. En effet, lors de la conception ou de la redéfinition d'un système de production, il est nécessaire d'étudier ses performances afin d'obtenir le système le plus adapté aux exigences telles que la capacité de production ou le délai de fabrication [Pujo and Kieffer, 2002].

Les systèmes flow-shop contrôlés ont ensuite été simulés selon un grand nombre de configurations et de situations diverses. Les résultats mettent en évidence la criticité de l'activité du contrôleur et son influence sur la performance des procédés pilotés. Nous avons ainsi montré que le contrôleur central fixe une borne qui, si elle prédomine sur celle établie par la machine la plus lente, le positionne comme étant le goulot d'étranglement du système de production. Nous généralisons alors le calcul du délai optimal τ_{max} d'inter-sorties de pièces pour un système flow-shop pour l'étendre lorsqu'une ressource de contrôle est mise en œuvre.

L'utilisation de plusieurs règles de tri courantes pour ordonnancer l'activité du contrôleur montre qu'aucune ne permet d'atteindre de manière systématique la cadence optimale de production. Nous avons alors proposé une méthode d'évaluation des conditions d'atteignabilité du délai optimal τ_{max} d'inter-sorties de pièces et montré que la

configuration de la gamme opératoire ne permet pas toujours à un système flow-shop contrôlé de produire à la cadence optimale. Nous avons ensuite dérivé de cette méthode une technique de synchronisation des machines autorisant l'atteinte de la meilleure cadence en fonction de la configuration de gamme opératoire. Il s'agit alors de mettre en application un ordonnancement hors ligne de l'activité de toutes les ressources du système afin de maîtriser les accès concurrents de messages de requête soumis au contrôleur central. La technique par synchronisation des machines impose néanmoins de répartir la fonction de coordination entre les machines de production et le contrôleur.

La centralisation de la fonction de coordination correspond à une caractéristique majeure du système de pilotage étudié et représente une donnée essentielle de la problématique posée. Nous avons alors poursuivi et orienté nos travaux vers l'étude de règles de tri plus spécifiquement adaptées au contexte de notre objectif initial en permettant un ordonnancement en ligne de l'activité du contrôleur. Les contraintes de temps sous-jacentes et la minimisation du retard algébrique maximal L_{max} à viser, nous ont amenés à étudier une adaptation des algorithmes *EDF* et *LLF* largement utilisés dans le domaine des systèmes temps réels. Le choix de ces algorithmes comme base de réflexion est motivé par leur optimalité démontrée dans le cas de problèmes d'ordonnancement de travaux à exécuter par une ressource unique et présentant des dates d'activation propres en utilisant le mécanisme de préemption. L'adaptation que nous proposons s'appuie sur la prise en compte des caractéristiques d'échéance et de laxité opératoires des machines de production pour évaluer le niveau d'urgence des messages de requête. Nous désignons les deux règles de tri par les acronymes *EoDF* et *LoLF* afin de rappeler les algorithmes sur lesquels elles sont basées et de préciser le contexte opératoire global considéré. Les résultats obtenus montrent que les règles de tri proposées permettent aux systèmes flow-shop contrôlés de produire à la meilleure cadence atteignable. De plus, l'exécution des systèmes flow-shop contrôlés à l'aide des règles de tri *EoDF* et *LoLF* se déroule en limitant le nombre de pièces en encours à tout instant sur le procédé.

Nos travaux formalisent l'impact d'un contrôleur central sur la performance d'un système flow-shop et plus particulièrement sur sa capacité à minimiser le makespan. Nous avons démontré que la mise en œuvre d'un contrôleur peut changer la position du goulot d'étranglement au sein d'un système de production en déplaçant la borne sur la cadence d'un équipement mécanique vers un équipement affecté au traitement d'informations. Nous avons alors apporté une solution efficace pour ordonnancer l'activité du contrôleur

central et répondre au mieux aux impératifs de productivité. Les résultats peuvent être étendus à d'autres contextes et sont, sous certaines conditions liées à la nature des travaux et des ressources, applicables à tout système flow-shop mettant en œuvre une ressource dont la disponibilité est partagée par l'ensemble des ressources implantées en série. Il est ainsi possible d'envisager leur application à un robot assurant des opérations pouvant être préemptées.

8.2 Travaux prospectifs

Un certain nombre de travaux prospectifs peuvent être envisagés. Tout d'abord, il paraît intéressant d'élargir l'étude à des systèmes flow-shop coordonnés par un contrôleur multi-processeurs assurant ainsi le traitement simultané de plusieurs messages de requête. Le choix des règles de tri basées sur les algorithmes *EDF* et *LLF* est en effet argumenté par leur niveau de performance en mode mono-processeur. Cependant, ces algorithmes ne sont pas optimaux dans un environnement multi-processeurs. Alors qu'il impose généralement un plus grand nombre de permutations de contexte que l'algorithme *EDF* en mode mono-processeur, ce qui explique qu'il soit moins utilisé, l'algorithme *LLF* présente certains avantages par rapport à *EDF* lorsque le système intègre plusieurs processeurs [Towsley and Panwar, 1990]. De plus, lorsqu'un jeu de tâches est ordonnançable dans un environnement multi-processeurs avec l'algorithme *EDF*, il est prouvé qu'il l'est également avec l'algorithme *LLF* [Dertouzos and Mok, 1989] [Nissanke, 1997]. D'autres règles de tri basées sur des algorithmes plus adaptés au mode multi-processeurs méritent en outre d'être étudiées. Les éléments de configuration du modèle sous forme d'un réseau de Petri que nous avons élaboré permettent la prise en compte de l'aspect multi-processeurs du contrôleur.

Dans le même ordre d'idées, les travaux peuvent aussi être étendus au cas des systèmes flow-shop hybrides contrôlés.

Un autre point à approfondir concerne le calcul de la cadence de production atteignable et la recherche de solutions de synchronisation de machines qui en découlent. Il s'agit d'évaluer la complexité du problème relatif à l'espace de combinaisons à explorer et de proposer un algorithme efficace permettant d'établir la meilleure solution ou une solution acceptable selon différents paramètres tels la minimisation des permutations de contexte ou la simple satisfaction d'un délai maximal d'inter-sorties de pièces.

Le contexte de variabilité de la durée des cycles opératoires successifs des machines est aussi un axe de réflexion et de poursuite des travaux. Le comportement des machines d'un système de production réel n'est en effet jamais totalement déterministe. Une telle considération peut engendrer une alternance de la position dans le système du goulot d'étranglement. Par conséquent, la prise en compte d'aléas de fonctionnement ou d'exécution fluctuante mais bornée des cycles opératoires présente un certain intérêt.

Enfin, la recherche d'un test de faisabilité des ordonnancements à partir des règles de tri que nous proposons apparaît comme un sujet d'étude qu'il serait intéressant d'explorer.

Bibliographie

- [Aggoune, 2003] Aggoune, R. (2003). *Minimizing the makespan for the flow shop scheduling problem with availability constraints*. European Journal of Operational Research, volume 153, numéro 3, pages 534–543.
- [Ahmadi and Bagchi, 1990] Ahmadi, R. and Bagchi, U. (1990). *Improved lower bounds for minimizing the sum of computation times of n jobs over m machines in a flow shop*. European Journal of Operational Research, volume 44, numéro 3, pages 331–336.
- [Aho et al., 1974] Aho, A., Hopcroft, J., and Ullman, J. (1974). *The design and analysis of computer algorithms*. Addison-Wesley.
- [Aléonard et al., 1994] Aléonard, C., Dindeleux, E., and Soënen, R. (1994). *Conduite des systèmes de production*. Techniques de l'Ingénieur.
- [Alla and David, 1992] Alla, H. and David, R. (1992). *Du Grafcet aux réseaux de Petri*. Hermès.
- [Allaoui and Artiba, 2004] Allaoui, H. and Artiba, A. (2004). *Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints*. Computers & Industrial Engineering, volume 47, numéro 4, pages 431–450.
- [Anderson, 1997] Anderson, E. (1997). *The management of manufacturing*. Addison-Wesley.
- [Anderson et al., 2005] Anderson, J., Bud, V., and Devi, U. (2005). *An EDF-based scheduling algorithm for multiprocessor soft real-time systems*. Proceedings of the 17th Euromicro Conference on Real-Time Systems, pages 199–208.
- [Archimède, 1991] Archimède, B. (1991). *Conception d'une architecture réactive distribuée et hiérarchisée pour le pilotage des systèmes de production*. Thèse de Doctorat, Université de Bordeaux 1.

- [Baccelli et al., 1992] Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and linearity : an algebra for discrete event systems*. John Wiley & Sons.
- [Baker, 1975] Baker, K. (1975). *A comparative survey of flowshop algorithms*. Operations Research, volume 23, pages 62–73.
- [Balbo et al., 1995] Balbo, G., Conte, G., Donatelli, S., Franceschinis, G., and Marsan, M. (1995). *Modelling with generalized stochastic petri nets*. John Wiley & Sons.
- [Banks, 1998] Banks, J. (1998). *Handbook of simulation : principles, methodology, advances, applications, and practices*. John Wiley & Sons.
- [Bansal, 1977] Bansal, S. (1977). *Minimizing the sum of completion times of n-jobs over M-machines in a flowshop – A branch and bound approach*. American Institute of Industrial Engineers Transactions, volume 9, numéro 3, pages 306–311.
- [Barad, 1998] Barad, M. (1998). *Timed Petri nets as a verification tool*. Proceedings of the 30th Winter Simulation Conference, pages 547–554.
- [Barton and Pantelides, 1994] Barton, P. and Pantelides, C. (1994). *Modeling of combined discrete/continuous processes*. American Institute of Chemical Engineers, volume 40, numéro 6, pages 966–979.
- [Baybars, 1986] Baybars, I. (1986). *A survey of exact algorithms for the simple assembly line balancing*. Management Science, volume 32, numéro 8, pages 909–932.
- [Baynat, 2000] Baynat, B. (2000). *Théorie des files d'attente*. Hermès.
- [Becker and Scholl, 2006] Becker, C. and Scholl, A. (2006). *A survey on problems and methods in generalized assembly line balancing*. European Journal of Operational Research, volume 168, numéro 3, pages 694–715.
- [Bel and Dubois, 1985] Bel, G. and Dubois, D. (1985). *Modélisation et simulation de systèmes automatisés de production*. Automatique Productique Informatique Industrielle, volume 19, numéro 1, pages 3–43.
- [Bellman, 1957] Bellman, R. (1957). *Dynamic programming*. Princeton University Press.
- [Berry, 1993] Berry, G. (1993). *Preemption in concurrent systems*. Proceedings of the 13th Conference on Foundations of Software Technology and Theoretical Computer Science, pages 72–93.

-
- [Bestwick and Lockger, 1979] Bestwick, P. and Lockger, K. (1979). *A practical approach to production scheduling*. International Journal of Production Research, volume 17, numéro 2, pages 95–110.
- [Blackstone et al., 1982] Blackstone, J., Philips, D., and Hogg, G. (1982). *A state of the art survey of dispatching rules for manufacturing job shop tasks*. International Journal of Production Research, volume 20, numéro 1, pages 27–45.
- [Blazewicz, 1987] Blazewicz, J. (1987). *Selected topics in scheduling theory*. Annals of Discrete Mathematics, volume 31, pages 1–60.
- [Blazewicz et al., 1996] Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., and Weglarz, J. (1996). *Scheduling computer and manufacturing processes*. Springer.
- [Bouchentouf, 1991] Bouchentouf, T. (1991). *Ordonnancement sous contraintes de précédence dans les systèmes temps réel*. Thèse de Doctorat, Université de Nantes.
- [Boucon, 1991] Boucon, D. (1991). *Ordonnancement d’atelier : aide au choix de règles de priorité*. Thèse de Doctorat, École Nationale Supérieure de l’Aéronautique et de l’Espace de Toulouse.
- [Bouquard et al., 2005] Bouquard, J., Billaut, J., Kubzin, M., and V.A.Strusevich (2005). *Two-machine flow shop scheduling problems with no-wait jobs*. Operations Research Letters, volume 33, numéro 3, pages 255–262.
- [Boysen et al., 2006] Boysen, N., Fliedner, M., and Scholl, A. (2006). *Assembly line balancing : which model to use when ?* Rapport technique, Wirtschaftswissenschaftlichen Fakultät, Friedrich-Schiller-Universität Jena.
- [Brams, 1983] Brams, G. (1983). *Réseaux de Petri : théorie et pratique*. Masson.
- [Breit, 2004] Breit, J. (2004). *A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint*. Computers & Operations Research, volume 33, numéro 8, pages 2143–2153.
- [Brilman, 1995] Brilman, J. (1995). *L’entreprise réinventée*. Les Éditions d’Organisation.
- [Brown et al., 1984] Brown, J., Dubois, D., Rathmill, K., Sethi, S., and Stecke, K. (1984). *Classification of flexible manufacturing systems*. The FMS magazine, volume 2, numéro 2, pages 114–117.

- [Brucker et al., 2003] Brucker, P., Heitmann, S., and Hurink, J. (2003). *Flow-shop problems with intermediate buffers*. Operations Research Spectrum, volume 25, numéro 4, pages 549–574.
- [Brucker and Knust, 2006] Brucker, P. and Knust, S. (2006). *Complexity results for scheduling problems*. Rapport technique, Institut für Mathematik, Universität Osnabrück.
- [Brucker et al., 2005] Brucker, P., Knust, S., and Wang, G. (2005). *Complexity results for flow-shop problems with a single server*. European Journal of Operational Research, volume 165, numéro 2, pages 398–407.
- [Buttazzo, 1997] Buttazzo, G. (1997). *Hard real-time computing systems*. Kluwer.
- [Buxey et al., 1973] Buxey, G., Slack, N., and Wild, R. (1973). *Production flow line system design : a review*. American Institute of Industrial Engineers Transactions, volume 5, numéro 1, pages 37–48.
- [Buzacott and Shanthikumar, 1993] Buzacott, J. and Shanthikumar, J. (1993). *Stochastic models of manufacturing*. Prentice-Hall.
- [Calvez, 1990] Calvez, J. (1990). *Spécification et conception des systèmes*. Masson.
- [Campbell et al., 1970] Campbell, H., Dudek, R., and Smith, M. (1970). *An heuristic algorithm for the n jobs, m machines sequencing problem*. Management Science, volume 16, numéro 10, pages 630–637.
- [Carlier and Chétienne, 1988] Carlier, J. and Chétienne, P. (1988). *Problème d'ordonnement : modélisation, algorithmes et complexité*. Masson.
- [Carlier and Rebaÿ, 1996] Carlier, J. and Rebaÿ, I. (1996). *Two branch and bound algorithms for the permutation flow shop problem*. European Journal of Operational Research, volume 90, numéro 2, pages 238–251.
- [Cassandra, 1993] Cassandra, C. (1993). *Discrete event systems : modeling and performance analysis*. Aksen Associates Incorporated Publishers.
- [Cassandra et al., 2006] Cassandra, C., Giua, A., Seatzu, C., and Zaytoon, J. (2006). *Analysis and design of hybrid systems*. Elsevier.
- [Cassandra et al., 1995] Cassandra, C., Lafortune, S., and Olsder, G. (1995). *Introduction to the modelling, control and optimization of discrete event systems*, volume Trends in control : a european perspective, pages 217–292. Springer, A. Isidori edition.

-
- [Castagna, 2004] Castagna, P. (2004). *Contribution \tilde{A} la modélisation, la simulation et la commande de systèmes de production et de transitique*. Habilitation à Diriger des Recherches, Université de Nantes.
- [Cernault, 1988] Cernault, A. (1988). *La simulation des systèmes de production*. Cépaduès.
- [Chappe and Bourjault, 1988] Chappe, D. and Bourjault, A. (1988). *Utilisation des réseaux de Petri temporisés pour la conception et l'évaluation des systèmes d'assemblage automatisés*. Proceedings of the 12th International Association for Mathematics and Computers in Simulation World Conference, pages 538–542.
- [Cheng and Liu, 2003] Cheng, T. and Liu, Z. (2003). *Approximability of two-machine no-wait flowshop scheduling with availability constraints*. Operations Research Letters, volume 31, numéro 4, pages 319–322.
- [Chetto, 1990] Chetto, H. (1990). *L'ordonnancement dans les systèmes de contrôle temps réel à contraintes strictes*. Thèse de Doctorat, Université de Nantes.
- [Chetto and Chetto, 1989] Chetto, H. and Chetto, M. (1989). *Some results of the earliest deadline scheduling algorithm*. IEEE Transactions on Software Engineering, volume 15, numéro 10, pages 1161–1169.
- [Chetty and Gnanasekaran, 1996] Chetty, O. and Gnanasekaran, O. (1996). *Modelling, simulation and scheduling flexible assembly systems with coloured Petri nets*. Journal of Advanced Manufacturing Technology, volume 11, numéro 6, pages 430–438.
- [Chevalier, 1977] Chevalier, A. (1977). *La programmation dynamique*. Dunod.
- [Cho et al., 2002] Cho, S., Lee, S., and et K.J. Lin, A. H. (2002). *Efficient real-time scheduling algorithms for multiprocessor systems*. Transaction, volume E85-B, numéro 12, pages 2859–2867.
- [Cho and Sahni, 1981] Cho, Y. and Sahni, S. (1981). *Preemptive scheduling of independent jobs with release and due times on open, flow and job shops*. Operations Research, volume 29, numéro 3, pages 511–522.
- [Chong et al., 2005] Chong, C., Low, M., Sivakumar, A., and Gay, K. (2005). *Using simulation based approach to improve on the mean cycle time performance of dispatching rules*. Proceedings of the 37th Winter Simulation Conference, pages 2194–2202.

- [Christopher et al., 2005] Christopher, J., Kuhl, M., and Hirschman, K. (2005). *Simulation analysis of dispatching rules for automated material handling systems and processing tools in semiconductor fabs*. Proceedings of the IEEE International Symposium on Semiconductor Manufacturing, pages 84–87.
- [Chu and Proth, 1996] Chu, C. and Proth, J. (1996). *L'ordonnancement et ses applications*. Masson.
- [Chu et al., 1995] Chu, C., Proth, J., and Sethi, S. (1995). *Heuristic procedures for minimizing makespan and the number of required pallets in flow shops*. European Journal of Operational Research, volume 86, numéro 3, pages 491–502.
- [Cohen et al., 1983] Cohen, G., Dubois, D., Quadrat, J., and Viot, M. (1983). *Analyse du comportement périodique de systèmes de production par la théorie des dioÃdes*. Rapport technique RR-0191, Institut National de Recherche en Informatique et Automatique.
- [Colin, 2003] Colin, R. (2003). *Ordonnancement et pilotage d'atelier*. AFNOR.
- [Colombo et al., 1994] Colombo, A., Martinez, J., and Carelli, R. (1994). *Formal specification of complex production systems using coloured Petri nets*. Proceedings of the IEEE International Conference on Robotics and Automation, pages 1713–1718.
- [Commault and Dallery, 1990] Commault, C. and Dallery, Y. (1990). *Production rate of transfer lines without buffer storage*. IEEE Transactions, volume 22, numéro 4, pages 315–329.
- [Conway et al., 1967] Conway, R., Maxwell, W., and Miller, L. (1967). *Theory of scheduling*. Addison-Wesley.
- [Cottet et al., 1999] Cottet, F., Delacroix, J., Kaiser, C., and Mammeri, Z. (1999). *Ordonnancement temps réel : ordonnancement centralisé*. Techniques de l'Ingénieur.
- [Cottet et al., 2000] Cottet, F., Delacroix, J., Kaiser, C., and Mammeri, Z. (2000). *Ordonnancement temps réel*. Hermès.
- [Dang, 2000] Dang, T. (2000). *Vérification et synthèse des systèmes hybrides*. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Davis and Kenet, 1993] Davis, J. and Kenet, J. (1993). *Single machine scheduling with early and tardy completion costs*. Naval Research Logistics Quarterly, volume 40, numéro 1, pages 85–101.

-
- [Delacroix, 1994] Delacroix, J. (1994). *Stabilité et régisseur d'ordonnancement en temps réel*. Technique et Science Informatiques, volume 13, numéro 3, pages 223–250.
- [Denis and Lesage, 1995] Denis, B. and Lesage, J. (1995). *Formalisation de la conception d'architecture de conduite des systèmes de production*. Proceedings of the 2nd International Conference on Industrial Automation, pages 257–264.
- [Dertouzos, 1974] Dertouzos, M. (1974). *Control robotics : the procedural control of physical processors*. Proceedings of the International Federation for Information Processing Congress, pages 807–813.
- [Dertouzos and Mok, 1989] Dertouzos, M. and Mok, A. (1989). *Multiprocessor on-line scheduling of hard real time tasks*. IEEE Transactions on Software Engineering, volume 15, numéro 12, pages 1497–1506.
- [Desrochers and Al-Jaar, 1995] Desrochers, A. and Al-Jaar, R. (1995). *Application of Petri nets in manufacturing systems*. IEEE Press.
- [Dhall, 1977] Dhall, S. (1977). *Scheduling periodic-time critical jobs on a single processor and multiprocessor computing systems*. Thèse de Doctorat, Université de l'Illinois.
- [DiCesare et al., 1993] DiCesare, F., Harhalakis, G., Proth, J., Silva, M., and Verdant, F. (1993). *Practise of Petri nets in manufacturing*. Chapman & Hall.
- [Dilts et al., 1991] Dilts, D., Boyd, N., and Whorms, H. (1991). *The evolution of control architectures for automated manufacturing systems*. Journal of Manufacturing Systems, volume 10, numéro 1, pages 79–93.
- [Dindeleux, 1992] Dindeleux, E. (1992). *Proposition d'un modèle et d'un système interactif d'aide \tilde{A} la décision pour la conduite d'atelier*. Thèse de Doctorat, Université de Valenciennes.
- [Donath and Graves, 1988] Donath, M. and Graves, R. (1988). *Flexible assembly systems : an approach for near real-time scheduling and routing of multiple products*. International Journal of Production Research, volume 26, numéro 12, pages 1903–1919.
- [Drake and Smith, 1996] Drake, G. and Smith, J. (1996). *Simulation system for real-time planning, scheduling and control*. Proceedings of the 28th Winter Simulation Conference, pages 1083–1090.

- [Du and Leung, 1993] Du, J. and Leung, J. (1993). *Minimizing mean flow time in two-machine open shops and flow shops*. Journal of Algorithms, volume 14, numéro 1, pages 24–44.
- [Dudek et al., 1991] Dudek, R., Panwalkar, S., and Smith, M. (1991). *The lessons of flowshop scheduling research*. Operations Research, volume 40, numéro 1, pages 7–13.
- [Dupas, 2004] Dupas, R. (2004). *Amélioration de performance des systèmes de production : apport des algorithmes évolutionnistes aux problèmes d’ordonnancement cycliques et flexibles*. Habilitation à Diriger des Recherches, Université d’Artois.
- [El-Tamini et al., 1989] El-Tamini, A., Suliman, S., and Williams, D. (1989). *A simulation study of part sequencing in a flexible assembly cell*. International Journal of Production Research, volume 27, numéro 10, pages 1769–1793.
- [Eldabi et al., 2002] Eldabi, T., Irani, Z., Paul, R., and Love, P. (2002). *Quantitative and qualitative decision-making methods in simulation modelling*. Management Decision, volume 40, numéro 1, pages 64–73.
- [Emmons and Mathur, 1995] Emmons, H. and Mathur, K. (1995). *Lot sizing in a no-wait flow shop*. Operations Research Letters, volume 17, numéro 4, pages 159–184.
- [Erickson et al., 1987] Erickson, C., Vandenberg, A., and Miles, T. (1987). *Simulation, animation and shop-floor control*. Proceedings of the 19th Winter Simulation Conference, pages 649–652.
- [Erschler et al., 1992] Erschler, J., Fontan, G., and Roubellat, F. (1992). *Ordonnancement en ateliers spécialisés*. Vuibert.
- [Espinouse, 1998] Espinouse, M. (1998). *Flowshop et extensions : chevauchement des tâches, indisponibilité des machines et système de transport*. Thèse de Doctorat, Université de Grenoble.
- [Espinouse et al., 1999] Espinouse, M., Formanowicz, P., and Penz, B. (1999). *Minimizing the makespan in the two-machine no-wait flow-shop with limited machine availability*. Computers & Industrial Engineering, volume 37, numéro 1, pages 497–500.
- [Esquirol and Lopez, 1999] Esquirol, P. and Lopez, P. (1999). *L’ordonnancement*. Économica.

-
- [Esswein, 2003] Esswein, C. (2003). *Un apport de flexibilité pour l'ordonnancement robuste*. Thèse de Doctorat, Université de Tours.
- [Färber and Coves, 2005] Färber, G. and Coves, A. (2005). *Overview on sequencing in mixed model flowshop production line with static and dynamic context*. Rapport technique, Institut d'Organització i Control de Sistemes Industrials, Universitat Politècnica de Catalunya.
- [Ferrell et al., 2000] Ferrell, W., Sale, J., Sams, J., and Yellamraju, M. (2000). *Evaluating simple scheduling rules in a mixed shop environment*. Computers & Industrial Engineering, volume 38, numéro 1, pages 39–66.
- [Ferrier and Boimond, 2004] Ferrier, J. and Boimond, J. (2004). *Systèmes dynamiques \tilde{A} événements discrets : du modèle \tilde{A} la commande*. Rapport technique, Laboratoire d'Ingénierie des Systèmes Automatisés, Université d'Angers.
- [Finke et al., 2002] Finke, G., Espinouse, M., and Jiang, H. (2002). *General flowshop models : job dependent capacities, job overlapping and deterioration*. International Transactions in Operational Research, volume 9, numéro 4, pages 399–415.
- [Fondrevelle, 2005] Fondrevelle, J. (2005). *Résolution exacte de problèmes d'ordonnancement de type flowshop de permutation en présence de contraintes d'écart temporels entre opérations*. Thèse de Doctorat, Institut National Polytechnique de Lorraine.
- [French, 1982] French, S. (1982). *Sequencing and scheduling : an introduction to the mathematics of the job-shop*. John Wiley & Sons.
- [Gardarin and Gardarin, 1996] Gardarin, G. and Gardarin, O. (1996). *Le client-serveur*. Eyrolles.
- [Garey and Johnson, 1979] Garey, M. and Johnson, D. (1979). *Computers and intractability : a guide to the theory of NP-completeness*. Freeman and Company.
- [Garey et al., 1976] Garey, M., Johnson, D., and Sethi, R. (1976). *The complexity of flowshop and jobshop scheduling*. Mathematical and Operations Research, volume 1, numéro 2, pages 117–129.
- [Gaubert, 1992] Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioïdes*. Thèse de Doctorat, École des Mines de Paris.

- [Gaujál et al., 1993] Gaujal, B., Jafari, M., and Gursoy, M. (1993). *Performance analysis of timed petri nets*. Proceedings of the International Conference on Systems, Man and Cybernetics, pages 312–316.
- [Gendreau, 1991] Gendreau, D. (1991). *Génération automatique de procédures de pilotage d'une cellule automatisée de production*. Thèse de Doctorat, École Centrale de Paris.
- [German et al., 1995] German, R., Kelling, C., Zimmermann, A., and Hommel, G. (1995). *TimeNET - A toolkit for evaluating non-markovian stochastic Petri nets*. Performance Evaluation, volume 24, numéro 1–2, pages 69–87.
- [Gershwin, 1993] Gershwin, S. (1993). *Manufacturing systems engineering*. Prentice-Hall.
- [Gershwin and Dallery, 1992] Gershwin, S. and Dallery, Y. (1992). *Manufacturing flow line systems : a review of models and analytical results*. Queueing systems theory and applications, volume 12, numéro 1, pages 3–94.
- [Gharbi and Haouari, 2005] Gharbi, A. and Haouari, M. (2005). *Optimal parallel machines scheduling with availability constraints*. Discrete Applied Mathematics, volume 148, numéro 1, pages 63–87.
- [Gladky et al., 2004] Gladky, A., Shafransky, M., and Strusevich, V. (2004). *Flow shop scheduling problems under machine-dependent precedence constraints*. Journal of Combinatorial Optimization, volume 8, numéro 1, pages 13–28.
- [Glover, 1977] Glover, F. (1977). *Heuristic for integer programming using surrogate constraints*. Decision Sciences, volume 8, pages 156–166.
- [Glover, 1986] Glover, F. (1986). *Future paths for integer programming and links to artificial intelligence*. Computers & Operations Research, volume 13, numéro 5, pages 533–549.
- [Glover, 1989] Glover, F. (1989). *Tabu search, part 1*. ORSA Journal of Computing, volume 1, numéro 3, pages 190–206.
- [Glover, 1990] Glover, F. (1990). *Tabu search, part 2*. ORSA Journal of Computing, volume 2, numéro 1, pages 4–62.
- [Goldberg, 1989] Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley.

-
- [Goldratt and Cox, 1993] Goldratt, E. and Cox, J. (1993). *Le but : un processus de progrès permanent*. AFNOR.
- [Gonzalez and Sahni, 1976] Gonzalez, T. and Sahni, S. (1976). *Openshop scheduling to minimize finish time*. Journal of the Association for Computing Machinery, volume 23, numéro 4, pages 665–679.
- [Gonzalez and Sahni, 1978] Gonzalez, T. and Sahni, S. (1978). *Flowshop and jobshop schedules : complexity and approximation*. Operations Research, volume 26, numéro 1, pages 36–52.
- [Grabot, 2006] Grabot, B. (2006). *Ordonnancement d'ateliers manufacturiers*. Techniques de l'Ingénieur.
- [Grabot et al., 1993] Grabot, B., Geneste, L., and Dupeux, A. (1993). *Ordonnancement réactif par utilisation de règles floues paramétrables*. Revue d'Automatique et de Productique Appliquées, volume 6, numéro 3, pages 273–290.
- [Graham et al., 1979] Graham, R., Lawler, E., Lenstra, J., and Rinnooy Kan, A. (1979). *Optimization and approximation in deterministic sequencing and scheduling : a survey*. Annals of Discrete Mathematics, volume 5, pages 287–326.
- [Grieco et al., 2001] Grieco, A., Semeraro, Q., and Tolio, T. (2001). *A review of different approaches to the FMS loading problem*. The International Journal of Flexible manufacturing Systems, volume 13, numéro 4, pages 361–384.
- [Groover, 1996] Groover, M. (1996). *Fundamentals of modern manufacturing : materials, processes and systems*. Prentice-Hall.
- [Guinet and Legrand, 1998] Guinet, A. and Legrand, M. (1998). *Reduction of job-shop problems to flow-shop problems with precedence constraints*. European Journal of Operational Research, volume 109, numéro 1, pages 96–110.
- [Gupta, 1972] Gupta, J. (1972). *Heuristic algorithms for the multistage flow shop problem*. American Institute of Industrial Engineers Transactions, volume 4, numéro 1, pages 11–18.
- [Gupta, 1979] Gupta, J. (1979). *A review of flow-shop scheduling research*. Martinus Nijhoff.
- [Gupta and Stafford, 2006] Gupta, J. and Stafford, E. (2006). *Flowshop scheduling re-*

- search after five decades*. European Journal of Operational Research, volume 169, numéro 3, pages 699–711.
- [Habchi, 2001] Habchi, G. (2001). *Conceptualisation et modélisation pour la simulation et des systèmes de production*. Habilitation à Diriger des Recherches, Université de Savoie.
- [Hall et al., 2000] Hall, N., Potts, C., and Srikandarajah, C. (2000). *Parallel machine scheduling with a common server*. Discrete Applied Mathematics, volume 102, numéro 3, pages 223–243.
- [Harley, 1984] Harley, J. (1984). *FMS at work*. North-Holland.
- [Haupt, 1989] Haupt, R. (1989). *A survey of priority rule-based scheduling*. Operation Research Spektrum, volume 11, numéro 1, pages 3–16.
- [Haverkort, 2000] Haverkort, B. (2000). *Makovian models for performance and dependability evaluation*. Proceedings of the 1st European Educational Summer School on Trends in Computer Sciences, pages 38–83.
- [Heiner, 1997] Heiner, M. (1997). *Verification and optimization of control programs by Petri nets without state explosion*. Proceedings of the International Conference on Application and Theory of Petri Nets, pages 69–84.
- [Herrmann, 2006] Herrmann, J. (2006). *Handbook of production scheduling*. Springer.
- [Hicks and Pongcharoen, 2006] Hicks, C. and Pongcharoen, P. (2006). *Dispatching rules for production scheduling in the capital goods industry*. International Journal of Production Economics, volume 104, numéro 1, pages 154–163.
- [Hildebrandt et al., 1999] Hildebrandt, J., Golasowski, F., and Timmermann, D. (1999). *Scheduling coprocessor for enhanced least-laxity-first scheduling in hard real-time systems*. Proceedings of the 11th Euromicro Conference on Real-Time Systems, pages 208–215.
- [Hindi and Fleszar, 2004] Hindi, K. and Fleszar, K. (2004). *A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem*. Computers & Industrial Engineering, volume 47, numéro 1, pages 91–101.
- [Ho, 1985] Ho, Y. (1985). *On the perturbation analysis of discrete-event dynamic systems*. Journal of Optimization Theory and Applications, volume 45, numéro 4, pages 535–545.

-
- [Holland, 1962] Holland, J. (1962). *Outline for a logical theory of adaptative systems*. Journal of the Association of Computing Machinery, volume 9, numéro 3, pages 297–314.
- [Holland, 1975] Holland, J. (1975). *Adaptation in natural an artificial systems*. University of Michigan Press.
- [Holliday and Vernon, 1985] Holliday, M. and Vernon, M. (1985). *A generalized timed Petri net model for performance evaluation*. Proceedings of the International Workshop on Timed Petri Nets, pages 181–190.
- [Holthaus and Rajendran, 1997] Holthaus, . and Rajendran, C. (1997). *Efficient dispatching rules for scheduling in a job-shop*. International Journal of Production Economics, volume 48, numéro 1, pages 87–105.
- [Hoogeveen, 2003] Hoogeveen, H. (2003). *Multicriteria scheduling*. Rapport technique, Departement of Computer Science, University of Utrecht.
- [Hopcroft and Ullman, 1979] Hopcroft, J. and Ullman, J. (1979). *Introduction to automata theory*. Addison-Wesley.
- [Hopp et al., 2005] Hopp, W., Irvani, S., and Shou, B. (2005). *Serial agile production systems with automation*. Operations Research, volume 53, numéro 5, pages 852–866.
- [Houari and Daouas, 1999] Houari, M. and Daouas, T. (1999). *Optimal scheduling of the 3-machine assembly-type flow shop*. RAIRO Recherche Opérationnelle, volume 33, numéro 4, pages 439–445.
- [Hunsucker and Shan, 1994] Hunsucker, J. and Shan, J. (1994). *Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment*. European Journal of Operational Research, volume 72, numéro 1, pages 102–114.
- [Hurink and Knust, 2001] Hurink, J. and Knust, S. (2001). *Makespan minimization for flow-shop problems with transportation times and a single robot*. Discrete Applied Mathematics, volume 112, numéro 1, pages 199–216.
- [Irvani and Teo, 2005] Irvani, S. and Teo, C. (2005). *Asymptotically optimal schedules for single-server flow shop with setup costs times*. Operations Research Letters, volume 33, numéro 4, pages 421–430.

- [Iyama and Ito, 1987] Iyama, T. and Ito, S. (1987). *The maximum production rate for an unbalanced multi-server flow line system with finite buffer storage*. International Journal Production Research, volume 25, numéro 8, pages 1157–1170.
- [Jackson, 1955] Jackson, J. (1955). *Scheduling a production line to minimize maximum tardiness*. Rapport technique, Management Science Research Project, University of California Los Angeles.
- [Jain, 1991] Jain, R. (1991). *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modelling*. John Wiley & Sons.
- [Jensen, 1992] Jensen, K. (1992). *Coloured Petri nets : basic concepts, analysis methods and practical use - Vol. 1 : Basic concepts*. Springer.
- [Jensen, 1994] Jensen, K. (1994). *Coloured Petri nets : basic concepts, analysis methods and practical use - Vol. 2 : Analysis methods*. Springer.
- [Jensen, 1997] Jensen, K. (1997). *Coloured Petri nets : basic concepts, analysis methods and practical use - Vol. 3 : Practical use*. Springer.
- [Jensen et al., 2007] Jensen, K., Kristensen, L., and Wells, L. (2007). *Coloured Petri nets and CPN Tools for modelling and validation of concurrent systems*. International Journal on Software Tools for Technology Transfer, volume 9, numéro 13, pages 213–254.
- [Jiang and Cheng, 2001] Jiang, F. and Cheng, A. (2001). *A context switch reduction technique for real-time task synchronization*. Proceedings of the 15th International Parallel and Distributed Processing Symposium, pages 31–31.
- [Jiang, 1997] Jiang, H. (1997). *Les problèmes d’ordonnancement avec durées opératoires variables*. Thèse de Doctorat, Université de Grenoble.
- [Johnson, 1954] Johnson, S. (1954). *Optimal two- and three-stage production schedules with set-up times included*. Naval Research Logistics Quarterly, volume 1, pages 61–68.
- [Jones et al., 1989] Jones, A., Barkmeyer, E., and David, W. (1989). *Issues in the design and implementation of a system architecture for computer integrated manufacturing*. Journal of Computer Integrated Manufacturing, volume 2, numéro 1, pages 65–76.
- [Juanole et al., 2000] Juanole, G., Abdellatif, S., and Gallon, L. (2000). *Nouveaux concepts sur les transitions du modèles RdPTS : Priorités dynamiques et attributs tem-*

-
- porels dynamiques*. Proceedings de la 1ère Conférence Internationale Francophone d'Automatique, pages 976–981.
- [Kamburowski, 2004] Kamburowski, J. (2004). *More on three-machine no-idle flow shops*. Computers & Industrial Engineering, volume 46, numéro 3, pages 461–466.
- [Kashyrskikh et al., 2001] Kashyrskikh, K., Potts, C., and Sevastianov, S. (2001). *A 3/2-approximation algorithm for two-machine flow-shop sequencing subject to release dates*. Discrete Applied Mathematics, volume 114, numéro 1, pages 255–271.
- [Kasturia et al., 1988] Kasturia, E., DiCesare, F., and Desrochers, A. (1988). *Real time control of multilevel manufacturing systems using coloured Petri nets*. Proceedings of the IEEE International Conference on Robotics and Automation, pages 1114–1119.
- [Kats and Levner, 2002] Kats, W. and Levner, E. (2002). *Cyclic scheduling in a robotic production line*. Journal of Scheduling, volume 5, numéro 1, pages 23–41.
- [Kirkpatrick et al., 1983] Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). *Optimization by simulating annealing*. Science, volume 220, pages 671–680.
- [Kizilişik, 1999] Kizilişik, O. (1999). *Predictive and reactive scheduling*. Rapport technique TR-06533, Department of Industrial Engineering, University of Ankara.
- [Kleinrock, 1975] Kleinrock, L. (1975). *Queueing systems. Vol 1 : Theory*. John Wiley & Sons.
- [Kohler, 1975] Kohler, W. (1975). *Exact, approximate and guaranteed accuracy algorithms for the flow-shop problem $n/2/F/F$* . Journal of the Association for Computing Machinery, volume 22, numéro 1, pages 106–114.
- [Kubiak et al., 1991] Kubiak, W., Sriskandarajah, C., and Zaras, K. (1991). *A note on the complexity of open shop scheduling problems*. Information Systems and Operational Research, volume 29, pages 284–294.
- [Kuhl and Laubisch, 2004] Kuhl, M. and Laubisch, G. (2004). *A simulation study of dispatching rules and rework strategies in semiconductor manufacturing*. Proceedings of the 15th Advanced Semiconductor Manufacturing Conference, pages 325–329.
- [Labetoulle, 1974] Labetoulle, J. (1974). *Un algorithme optimal pour la gestion des processus en temps réel*. Rapport technique RR-0073, Institut National de Recherche en Informatique et Automatique.

- [Lageweg et al., 1978] Lageweg, B., Lenstra, J., and Rinnooy Kan, A. (1978). *A general bounding scheme for the permutation flow-shop*. *Operations Research*, volume 26, numéro 1, pages 53–67.
- [Lamch, 2002] Lamch, D. (2002). *Verification and analysis of properties of dynamic systems based on Petri nets*. *Proceedings of the International Conference on Parallel Computing in Electrical Engineering*, pages 92–94.
- [Land and Doig, 1960] Land, A. and Doig, A. (1960). *An automatic method for solving discrete programming*. *Econometrica*, volume 28, pages 497–520.
- [Law and Kelton, 1991] Law, A. and Kelton, W. (1991). *Simulation, modeling and analysis*. Mac Graw Hill.
- [Law and Kelton, 2000] Law, A. and Kelton, W. (2000). *Simulation, modeling and analysis – 3rd edition*. Mac Graw Hill.
- [Le Parc et al., 1999] Le Parc, P., L’Her, D., Scharbarg, J., and Marcé, L. (1999). *Le Grafcet revisité à l’aide d’un langage synchrone flot de données*. *Technique et Science Informatiques*, volume 33, numéro 5-6, pages 651–683.
- [Lebre, 2006] Lebre, A. (2006). *Contrôle, ordonnancement et régulation des accès aux données persistantes dans les environnements multi-applicatifs haute performance*. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Lee et al., 1997] Lee, C., Lei, L., and Pinedo, M. (1997). *Current trends in deterministic scheduling*. *Annals of Operation Research*, volume 70, numéro 1, pages 1–41.
- [Leisten, 1990] Leisten, R. (1990). *Flow-shop problems with intermediate buffers*. *International Journal of Production Research*, volume 28, numéro 11, pages 2085–2100.
- [Lenstra et al., 1977] Lenstra, J., Rinnooy Kan, A., and Brucker, P. (1977). *Complexity of machine scheduling problems*. *Annals of Discrete Mathematics*, volume 1, pages 343–362.
- [Leung and Merrill, 1980] Leung, J. and Merrill, M. (1980). *A note on preemptive scheduling of periodic real-time tasks*. *Information Processing Letters*, volume 11, numéro 3, pages 115–118.
- [Lewis and Papadimitriou, 1981] Lewis, H. and Papadimitriou, C. (1981). *Elements of the theory of computation*. Prentice-Hall.

-
- [Li et al., 2004] Li, Y., Wang, F., and Lim, A. (2004). *A genetic algorithm for machine scheduling problem under shared resource constraints*. Computer Applications in Technology, volume 19, numéro 2, pages 77–83.
- [Liaw et al., 2005] Liaw, C., Cheng, C., and Chen, M. (2005). *Scheduling two-machine no-wait open shops to minimize makespan*. Computer and Operations Research, volume 32, numéro 4, pages 901–917.
- [Lim and Mac Mahon, 1994] Lim, C. and Mac Mahon, G. (1994). *The three-machine flow-shop problem with arbitrary precedence relations*. European Journal of Operational Research, volume 78, numéro 2, pages 216–223.
- [Lim, 1997] Lim, J. (1997). *A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line*. Computers & Industrial Engineering, volume 33, numéro 3, pages 789–792.
- [Linn and Zhang, 1999] Linn, R. and Zhang, W. (1999). *Hybrid flow shop scheduling : a survey*. Computers & Industrial Engineering, volume 37, numéro 1, pages 57–61.
- [Little and Hemmings, 1994] Little, D. and Hemmings, A. (1994). *Automated assembly scheduling : a review*. Computer Integrated Manufacturing Systems, volume 7, numéro 1, pages 51–61.
- [Little, 1961] Little, J. (1961). *A proof of the queueing formula $L = \lambda W$* . Operations Research, volume 9, pages 383–387.
- [Liu and Layland, 1973] Liu, C. and Layland, J. (1973). *Scheduling algorithms for multiprogramming in a hard-real-time environment*. Journal of the Association for Computing Machinery, volume 20, numéro 1, pages 46–61.
- [Liu and Mac Carthy, 1996] Liu, J. and Mac Carthy, B. (1996). *The classification of FMS scheduling problems*. International Journal of Production Research, volume 34, numéro 3, pages 647–656.
- [Liu, 1995] Liu, Z. (1995). *Performance analysis of stochastic timed Petri nets using linear programming approach*. Rapport technique RR-2642, Institut National de Recherche en Informatique et Automatique.
- [Lopez and Roubellat, 2001] Lopez, P. and Roubellat, F. (2001). *Ordonnancement de la production*. Hermès.

- [Mac Carthy and Liu, 1993] Mac Carthy, B. and Liu, J. (1993). *Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling*. International Journal in Production Research, volume 31, numéro 1, pages 59–79.
- [Mahajan and Ingalls, 2004] Mahajan, P. and Ingalls, R. (2004). *Evaluation of methods used to detect warm-up period in steady state simulation*. Proceedings of the 36th Winter Simulation Conference, pages 663–671.
- [Marsan et al., 1996] Marsan, M., Bobbio, A., and Donatelli, S. (1996). *Petri nets in performance analysis : an introduction*, pages 211–256. Lecture Notes in Computer Sciences. Springer.
- [Martinez et al., 1987] Martinez, J., Muro, P., and Silva, M. (1987). *Modelling, validation and software implementation of production systems using high level Petri nets*. Proceedings of the IEEE International Conference on Robotics and Automation, pages 1180–1185.
- [Masuda et al., 1985] Masuda, T., Ishi, H., and Nishida, T. (1985). *The mixed shop scheduling problem*. Discrete Applied Mathematics, volume 11, numéro 2, pages 175–186.
- [Mellor, 1966] Mellor, P. (1966). *A review of job shop scheduling*. Operational Research, volume 17, numéro 2, pages 161–171.
- [Meunier and Denis, 1997] Meunier, P. and Denis, B. (1997). *Validation du comportement dynamique des architectures de conduite des systèmes de production par simulation*. Proceedings de la 1ère Conférence Francophone de Modélisation et de Simulation, pages 229–238.
- [Mok, 1983] Mok, A. (1983). *Fundamental design problems of distributed systems for the hard real time environment*. Thèse de Doctorat, Massachusetts Institute of Technology.
- [Moncelet et al., 1998] Moncelet, G., Christensen, S., Demmou, H., Paludetto, M., and Porras, J. (1998). *Analysing a mechatronic system with coloured Petri nets*. International Journal on Software Tools and Technology Transfer, volume 2, numéro 2, pages 160–167.
- [Monsef, 1996] Monsef, Y. (1996). *Modélisation et simulation des systèmes complexes*. Lavoisier.

-
- [Moore, 1968] Moore, J. (1968). *An n -job, one machine sequencing algorithm for minimizing the number of late jobs*. *Management Science*, volume 15, numéro 1, pages 102–109.
- [Murata, 1989] Murata, T. (1989). *Petri nets : properties, analysis and applications*. *Proceedings of the IEEE*, volume 77, numéro 4, pages 541–580.
- [Ndjad and Zaytoon, 1999] Ndjad, C. and Zaytoon, J. (1999). *Synthèse de la ligne de commande \tilde{A} partir du Grafcet*. *Proceedings de la Conférence de Modélisation des Systèmes Réactifs*, pages 341–350.
- [Nilsson, 2003] Nilsson, K. (2003). *50 years of automobile occupant safety*. Autoliv Inc.
- [Nissanke, 1997] Nissanke, N. (1997). *Realtime systems*. Prentice Hall.
- [Oh and Yang, 1998] Oh, S. and Yang, S. (1998). *A modified least-laxity-first scheduling algorithm for real-time tasks*. *Proceedings of the 5th International Workshop on Real-Time Computing Systems and Applications*, pages 31–36.
- [Palmer, 1965] Palmer, D. (1965). *Sequencing jobs through a multi-stage process in the minimum total time – a quick method of obtaining a near optimum*. *Operations Research Quarterly*, volume 16, numéro 1, pages 101–107.
- [Panwalkar and Iskander, 1977] Panwalkar, S. and Iskander, W. (1977). *A survey of scheduling rules*. *Operations Research*, volume 25, numéro 1, pages 45–63.
- [Persson et al., 2006] Persson, A., Grimm, H., Ng, A., Lezama, T., Ekberg, J., Falk, S., and Stablum, P. (2006). *Simulation-based multi-objective optimization of a real-world scheduling problem*. *Proceedings of the 38th Winter Simulation Conference*, pages 1757–1764.
- [Petri, 1962] Petri, C. (1962). *Kommunikation mit automaten*. Rapport technique, Institut für Instrumentelle Mathematik, Universität Bonn.
- [Philips and Hunger, 1976] Philips, L. and Hunger, P. (1976). *Mathematical programming solution of a hoist scheduling problem*. *American Institute of Industrial Engineers Transactions*, volume 8, numéro 2, pages 219–225.
- [Piehler, 1960] Piehler, J. (1960). *Ein beitrag zum reinhenfolgeproblem*. *Unternehmensforschung*, volume 4, pages 138–142.
- [Pinedo, 1995] Pinedo, M. (1995). *Scheduling : theory, algorithms and systems*. Prentice-Hall.

- [Potts, 1985] Potts, C. (1985). *Analysis of heuristics for two-machine flow-shop sequencing subject to release dates*. Mathematical and Operations Research, volume 10, numéro 44, pages 576–584.
- [Potts et al., 1991] Potts, C., Shmoys, D., and Williamsons, D. (1991). *Permutation vs. non-permutation flow shop schedules*. Operations Research Letters, volume 10, numéro 5, pages 281–284.
- [Proth et al., 1993] Proth, J., Wang, L., and Xie, X. (1993). *A class of Petri nets for manufacturing systems*. Rapport technique RR-2055, Institut National de Recherche en Informatique et Automatique.
- [Proth and Xie, 1994] Proth, J. and Xie, X. (1994). *Les réseaux de Petri pour la conception et la gestion des systèmes de production*. Masson.
- [Pujo and Kieffer, 2002] Pujo, P. and Kieffer, J. (2002). *Méthodes du pilotage des systèmes de production*. Hermès.
- [Rajendran and Alicke, 2007] Rajendran, C. and Alicke, K. (2007). *Dispatching in flowshops with bottleneck machines*. Computer and Industrial Engineering, volume 52, numéro 1, pages 89–106.
- [Rajendran and Chandhuri, 1991] Rajendran, C. and Chandhuri, D. (1991). *An efficient heuristic approach to the scheduling of jobs in flowshop*. European Journal of Operational Research, volume 61, numéro 3, pages 318–325.
- [Rajendran and Holthaus, 1999] Rajendran, C. and Holthaus, O. (1999). *A comparative study of dispatching rules in dynamic flowshops and jobshops*. European Journal of Operational Research, volume 116, numéro 1, pages 156–170.
- [Ramchandani, 1973] Ramchandani, C. (1973). *Performance evaluation of asynchronous concurrent systems by timed Petri nets*. Thèse de Doctorat, Massachusetts Institute of Technology.
- [Ramesh and Cary, 1989] Ramesh, R. and Cary, J. (1989). *Multicriteria jobshop scheduling*. Computers & Industrial Engineering, volume 17, numéro 1, pages 597–602.
- [Ratzer et al., 2003] Ratzer, A., Wells, L., Lassen, H., Laursen, M., Qvortrup, J., Stissing, M., and S. Christensen, M. W., and Jensen, K. (2003). *CPN Tools for editing, simulating, and analysing coloured Petri nets*. Proceedings of the 24th International Conference on Applications and Theory of Petri Nets, pages 450–462.

-
- [Razouk and Phelps, 1984] Razouk, R. and Phelps, C. (1984). *Performance analysis using timed Petri nets*. Proceedings of the International Conference on Parallel Processing, pages 126–128.
- [Rebaine, 2005] Rebaine, D. (2005). *Flow shop vs. permutation shop with time delays*. Computers & Industrial Engineering, volume 48, numéro 2, pages 357–362.
- [Reddi and Sadowsky, 1976] Reddi, R. and Sadowsky, R. (1976). *A statistical comparison of the performance of simple dispatching rules for a particular set of job shops*. International Journal of Production Research, volume 14, numéro 1, pages 63–75.
- [Reddi and Ramamoorthy, 1972] Reddi, S. and Ramamoorthy, V. (1972). *On the flow-shop sequencing problem with no-wait process*. Operational Research Quaterly, volume 23, numéro 4, pages 323–331.
- [Rinnooy Kan, 1976] Rinnooy Kan, A. (1976). *Machine scheduling problems : classification, complexity and computations*. Martinus Nijhoff.
- [Rose, 2003] Rose, O. (2003). *Comparison of due-date dispatch rules in semiconductor manufacturing*. Proceedings of the International Engineering Research Conference, pages 18–23.
- [Rubino and Toutain, 1998] Rubino, G. and Toutain, L. (1998). *Réseaux locaux*. Techniques de l'Ingénieur.
- [Saadani, 2003] Saadani, N. (2003). *Ordonnancement des systèmes de production sans temps d'arrêt machine*. Thèse de Doctorat, Institut National des Sciences Appliquées de Lyon.
- [Saadani et al., 2005] Saadani, N., Guinet, A., and Moalla, M. (2005). *A travelling salesman approach to solve the $F/no - idle/C_{max}$ problem*. European Journal of Operational Research, volume 161, numéro 1, pages 11–20.
- [Sabuncuoglu and Karabük, 1999] Sabuncuoglu, I. and Karabük, S. (1999). *Rescheduling frequency in an FMS with uncertain processing times and unreliable machines*. Journal of Manufacturing Systems, volume 18, numéro 4, pages 268–283.
- [Sadfi, 2002] Sadfi, C. (2002). *Problèmes d'ordonnancement avec minimisation des encours*. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Sakarovitch, 1984] Sakarovitch, M. (1984). *Optimisation combinatoire : programmation discrète*. Freeman and Company.

- [Saraph, 2003] Saraph, P. (2003). *Shared resource capacity analysis in biotech manufacturing*. Proceedings of the 35th Winter Simulation Conference, pages 1247–1250.
- [Schild and Bussmann, 2007] Schild, K. and Bussmann, S. (2007). *Self-organization in manufacturing operations*. Communication of the Association for Computing Machinery, volume 50, numéro 12, pages 74–79.
- [Scudder and Hoffmann, 1985] Scudder, G. and Hoffmann, T. (1985). *An evaluation of value based dispatching rules in a flow shop*. European Journal of Operational Research, volume 22, numéro 3, pages 310–318.
- [Sculli, 1987] Sculli, D. (1987). *Priority dispatching rules in an assembly shop*. Omega, volume 15, numéro 1, pages 49–57.
- [Sculli and Tsang, 1990] Sculli, D. and Tsang, K. (1990). *Priority dispatching rules in an fabrication/assembly shop*. Mathematical and Computer Modelling, volume 13, numéro 3, pages 73–79.
- [Sethi and Sethi, 1990] Sethi, A. and Sethi, S. (1990). *Flexibility in manufacturing : a survey*. The International Journal of Flexible manufacturing Systems, volume 2, numéro 4, pages 289–328.
- [Shakhlevich et al., 1999] Shakhlevich, N., Sotskov, Y., and Werner, F. (1999). *Shop-scheduling problems with fixed and non-fixed machine orders of the jobs*. Annals of Operations Research, volume 92, numéro 1, pages 281–304.
- [Shtub and Dar-El, 1989] Shtub, A. and Dar-El, E. (1989). *A methodology for the selection of assembly systems*. The International Journal of Production Research, volume 27, numéro 11, pages 175–186.
- [Sifakis, 1977] Sifakis, J. (1977). *Use of Petri nets for performance evaluation*. Proceedings of the 3rd International Symposium on Measuring and Evaluating Computer Systems, pages 75–93.
- [Silva et al., 1998] Silva, M., Teruel, E., Valette, R., and Pingaud, H. (1998). *Petri nets and production systems*, pages 85–124. Lecture Notes in Computer Sciences. Springer.
- [Smith, 1956] Smith, W. (1956). *Various optimizers for single-stage production*. Naval Research Logistics Quarterly, volume 3, pages 59–66.
- [Snyder et al., 1995] Snyder, J., Whalley, D., and Parker, T. (1995). *Fast context*

-
- switches : compiler and architectural support for preemptive scheduling*. Microprocessors and Microsystems, volume 19, numéro 1, pages 35–42.
- [Sorenson, 1974] Sorenson, P. (1974). *A methodology for real-time system development*. Thèse de Doctorat, Université de Toronto.
- [Soukhal et al., 2005] Soukhal, A., Oulamara, A., and Martineau, P. (2005). *Complexity of flow shop scheduling problems with transportation constraints*. European Journal of Operational Research, volume 161, numéro 1, pages 32–41.
- [Spieksma and Woeginger, 2005] Spieksma, F. and Woeginger, G. (2005). *The no-wait flow-shop paradox*. Operations Research Letters, volume 33, numéro 6, pages 603–608.
- [Stankovic et al., 1985] Stankovic, J., Ramamritham, K., and Cheng, S. (1985). *Evaluation of a flexible task scheduling algorithm for distributed hard real-time systems*. IEEE Transactions on Computers, volume 34, numéro 12, pages 1130–1143.
- [Szwarc, 1983] Szwarc, W. (1983). *The flowshop problem with mean completion time criterion*. American Institute of Industrial Engineers Transactions, volume 15, numéro 1, pages 172–176.
- [Taillard, 1990] Taillard, E. (1990). *Some efficient heuristic methods for the flow shop scheduling problem*. European Journal of Operational Research, volume 47, numéro 1, pages 65–74.
- [Tanaev et al., 1994] Tanaev, V., Sotskov, Y., and Strusevich, V. (1994). *Scheduling theory : multi-stage systems*. Kluwer.
- [Tantawy et al., 1992] Tantawy, A., Tantawi, A., and Serpanos, D. (1992). *An adaptive scheduling scheme for dynamic service time allocation on a shared resource*. Proceedings of the International Conference on Distributed Computing Systems, pages 294–300.
- [Tchako, 1994] Tchako, J. (1994). *Contribution À la conception d'un système de pilotage distribué pour les systèmes automatisés de production*. Thèse de Doctorat, Université de Valenciennes.
- [Telle, 2003] Telle, O. (2003). *Gestion des chaînes logistiques dans le domaines aéronautique : aide À la conception au sein d'une relation donneur d'ordres fournisseur*. Thèse de Doctorat, École Nationale Supérieure de l'Aéronautique et de l'Espace.

- [Terracol and David, 1987] Terracol, C. and David, R. (1987). *Performances d'une ligne composée de machines et de stocks intermédiaires*. Automatique Productive Informatique Industrielle, volume 21, numéro 3, pages 239–262.
- [Thomesse, 2004] Thomesse, J. (2004). *Réseaux locaux industriels : concepts, typologie, caractéristiques*. Techniques de l'Ingénieur.
- [Thomesse, 2006] Thomesse, J. (2006). *Caractéristiques temporelles des réseaux industriels*. Techniques de l'Ingénieur.
- [T'kindt and Billaut, 2002] T'kindt, V. and Billaut, J. (2002). *Multicriteria scheduling : theory, models and algorithms*. Springer.
- [T'kindt et al., 2003] T'kindt, V., Gupta, J., and Billaut, J. (2003). *Two-machine flow-shop scheduling with a secondary resource criterion*. Computers & Operations Research, volume 30, numéro 4, pages 505–526.
- [Towsley and Panwar, 1990] Towsley, D. and Panwar, S. (1990). *On the optimality of the minimum laxity and earliest deadline scheduling for realtime multiprocessors*. Proceedings of the Euromicro Workshop on Real-Time, pages 17–24.
- [Trentesaux, 1996] Trentesaux, D. (1996). *Conception d'un système de pilotage distribué, supervisé et multicritère pour les systèmes automatisés de production*. Thèse de Doctorat, Institut National Polytechnique de Grenoble.
- [Trentesaux and Sénéchal, 2002] Trentesaux, D. and Sénéchal, O. (2002). *Conduite des systèmes de production manufacturière*. Techniques de l'Ingénieur.
- [Tsinarakis et al., 2005] Tsinarakis, G., Tsourveloudis, N., and Valavanis, K. (2005). *Petri net modeling of routing and operation flexibility in production systems*. Proceedings of the 13th Mediterranean Conference on Control and Automation, pages 352–357.
- [Umble and Srikanth, 1990] Umble, M. and Srikanth, M. (1990). *Synchronous manufacturing*. South Western.
- [Uthaisombut, 2004] Uthaisombut, P. (2004). *Generalization of EDF and LLF : identifying all optimal online algorithms for minimizing maximum lateness*. Proceedings of the 9th Scandinavian Workshop on Algorithm Theory, pages 420–430.
- [van der Aalst, 1992] van der Aalst, W. (1992). *Timed colored Petri Nets and their application to logistics*. Thèse de Doctorat, Technische Universiteit Eindhoven.

-
- [van der Aalst, 1994a] van der Aalst, W. (1994a). *Modelling and analysis of production systems using a Petri nets based approach*. Proceedings of the Conference on Computer Integrated Manufacturing in the Process Industries, pages 179–193.
- [van der Aalst, 1994b] van der Aalst, W. (1994b). *Using interval timed coloured Petri nets to calculate performance bounds*. Proceedings of the 7th International Conference of Modelling Techniques and Tools for Computer Performance Evaluation, pages 425–444.
- [Veeramani, 1994] Veeramani, D. (1994). *Control of manufacturing systems*. John Wiley & Sons.
- [Vignier, 1997] Vignier, A. (1997). *Contribution À la résolution de problèmes d’ordonnancement de type monogramme, multimachines (flowshop hybride)*. Thèse de Doctorat, Université de Tours.
- [Vignier et al., 1999] Vignier, A., Billaut, J., and Proust, C. (1999). *Les problèmes d’ordonnancement de type flow-shop hybride : état de l’art*. RAIRO Recherche Opérationnelle, volume 33, numéro 2, pages 117–183.
- [Viswanadham and Narahari, 1987] Viswanadham, N. and Narahari, Y. (1987). *Coloured Petri nets models for automated manufacturing systems*. Proceedings of IEEE the International Conference on Robotics and Automation, pages 1985–1990.
- [Wang et al., 2006] Wang, L., Zhang, L., and Zheng, D. (2006). *An effective hybrid genetic algorithm for flow shop scheduling with limited buffer*. Computers & Operations Research, volume 33, numéro 10, pages 2960–2971.
- [Wang et al., 2005] Wang, Z., Xing, W., and Bai, F. (2005). *No-wait flexible flowshop scheduling with no-idle machines*. Operations Research Letters, volume 33, numéro 6, pages 609–614.
- [Watson et al., 1999] Watson, J., Barbulescu, L., Howe, A., and Whitley, L. (1999). *Algorithm performance and problem structure for flow-shop scheduling*. Proceedings of the 16th National Conference on Artificial Intelligence, pages 688–695.
- [Watson et al., 2002] Watson, J., Barbulescu, L., Howe, A., and Whitley, L. (2002). *Contrsting structured and random permutation flow-shop scheduling problems : search-space topology and algorithm performance*. INFORMS Journal on Computing, volume 14, numéro 2, pages 98–123.

- [Welch, 1983a] Welch, P. (1983a). *A graphical approach to the initial transient problem in steady state simulation*. Proceedings of the 10th World Congress on System Simulation and Scientific Computation, pages 219–221.
- [Welch, 1983b] Welch, P. (1983b). *The statistical analysis of simulation results*. Computer Performance Modeling Handbook, pages 267–329. Lavenberg.
- [Wells, 2002] Wells, L. (2002). *Performance analysis using Petri nets*. Thèse de Doctorat, Aarhus Universitet.
- [Wells, 2006] Wells, L. (2006). *Performance analysis using CPN Tools*. Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools.
- [Widmer, 1991] Widmer, M. (1991). *Modèles mathématiques pour une gestion efficace des ateliers flexibles*. Presses Polytechniques Romandes.
- [Widmer and Hertz, 1989] Widmer, M. and Hertz, A. (1989). *A new heuristic method for the flow shop sequencing problem*. European Journal of Operational Research, volume 41, numéro 2, pages 186–193.
- [Wild, 1972] Wild, R. (1972). *Mass-production management. The design and operation of production flow-line systems*. John Wiley & Sons.
- [Wilf, 1989] Wilf, H. (1989). *Algorithmes et complexité*. Masson.
- [Xuong, 1992] Xuong, N. (1992). *Mathématiques discrètes et informatique*. Masson.
- [Yamazaki et al., 1985] Yamazaki, G., Kawashima, T., and Sakasekawa, H. (1985). *Reversibility of tandem blocking queueing systems*. Management Science, volume 31, numéro 1, pages 78–83.
- [Yamazaki et al., 1978] Yamazaki, G., Sakasekawa, H., and Kawashima, T. (1978). *Production rate estimated with flow-shop reversibility*. Bulletin of the Japan Society of Mechanical Engineering, volume 21, pages 167–171.
- [Yamazaki et al., 1992] Yamazaki, G., Sakasekawa, H., and Shanthikumar, J. (1992). *An optimal arrangement of stations in a tandem queueing systems with blocking*. Management Science, volume 38, numéro 1, pages 137–153.
- [Yates, 1983] Yates, B. (1983). *The decline and fall of the american automobile industry*. Vintage Books.

-
- [Zaytoon, 2001] Zaytoon, J. (2001). *Systèmes dynamiques hybrides*. Hermès.
- [Zaytoon et al., 1997] Zaytoon, J., Lesage, J., Marcé, L., Faure, J., and Lhoste, P. (1997). *Vérification et validation du Grafcet*. Journal Européen des Systèmes Automatisés, volume 31, numéro 4, pages 713–740.
- [Zhou and Jeng, 1998] Zhou, M. and Jeng, M. (1998). *Modeling, analysis, simulation, scheduling and control of semiconductor manufacturing systems : a Petri net approach*. IEEE Transactions on Semiconductor Manufacturing, volume 11, numéro 3, pages 333–357.
- [Zhou and Venkatesh, 1999] Zhou, M. and Venkatesh, K. (1999). *Modeling, simulation, and control of flexible manufacturing systems : a Petri net approach*. World Scientific.
- [Zhou and Petrov, 2006] Zhou, X. and Petrov, P. (2006). *Rapid and low-cost context-switch through embedded processor customization for real-time and control applications*. Proceedings of the 43rd Annual Conference on Design Automation, pages 352–357.
- [Zimmermann et al., 1996] Zimmermann, A., Bode, S., and Hommel, G. (1996). *Performance and dependability evaluation of manufacturing systems using Petri nets*. Proceedings of the 1st Workshop on Manufacturing Systems and Petri Nets, pages 235–250.
- [Zimmermann et al., 2000] Zimmermann, A., Freiheit, J., German, R., and Hommel, G. (2000). *Petri net modelling and performability evaluation with TimeNET 3.0*. Proceedings of the 11th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, pages 188–202.
- [Zimmermann et al., 2001] Zimmermann, A., Freiheit, J., and Huck, A. (2001). *A Petri net based engine for manufacturing systems*. International Journal of Production Research, volume 39, numéro 2, pages 225–253.
- [Zimmermann and Hommel, 1999] Zimmermann, A. and Hommel, G. (1999). *Modelling and evaluation of manufacturing systems using dedicated Petri nets*. Journal of Advanced Manufacturing Technology, volume 15, numéro 2, pages 132–137.
- [Zuberek, 1991] Zuberek, W. (1991). *Timed Petri nets : definition, properties, and applications*. Microelectronics and Reliability, volume 31, numéro 4, pages 627–644.

Symboles et notations

\mathcal{M}	Ensemble des machines de production implantées sur le système flow-shop.
k	Nombre de machines de production implantées sur le système flow-shop.
M_k	Identifiant de la dernière machine de production du système flow-shop.
M_i	Identifiant d'une machine de production donnée du système flow-shop.
$M_\circ(t)$	Identifiant de la machine de production la plus prioritaire du système flow-shop à l'instant t .
C	Contrôleur central du système flow-shop
\mathcal{J}	Ensemble des travaux à réaliser, des pièces à produire.
n	Nombre de travaux à réaliser, des pièces à produire. Taille du lot de production.
J_n	Identifiant du dernier travail à réaliser, de la dernière pièce, à produire.
J_j	Identifiant d'un travail donné, d'une pièce donnée, du lot de production
m_i	Nombre d'opérations affectées à la machine de production M_i .
\mathcal{O}_i	Ensemble des opérations, partie de la gamme de fabrication, affecté à la machine de production M_i pour la réalisation d'un travail, d'une pièce.
O_{m_i}	Identifiant de la dernière opération affectée à la machine de production M_i .

θ_i	Durée minimale nécessaire à la réalisation d'un travail sur la machine de production M_i . Durée effective de l'exécution de la séquence opératoire affectée à la machine de production M_i .
$\theta_{i,q}$	Durée nécessaire à l'exécution de l'opération $O_{i,q}$ assignée à la machine de production M_i et exécutée localement (opération mécanique).
$\theta_{i,q}^*$	Durée nécessaire à l'exécution de l'opération $O_{i,q}^*$ assignée à la machine de production M_i et exécutée à distance par le contrôleur central (traitement d'un message de requête).
$\overline{\theta}_i$	Délai maximal s'écoulant entre la fin d'un travail et le début du suivant pour la machine de production M_i . Latence d'activité maximale de la machine de production M_i .
θ_i^*	Délai nécessaire au traitement des messages de requête effectué par le contrôleur central pour la réalisation d'un travail, d'une pièce, par la machine de production M_i .
$\psi_{i-1,i}$	Délai d'acheminement d'une pièce entre deux machines de production M_{i-1} et M_i implantées consécutivement sur le système flow-shop.
$\varpi_{i-1,i}$	Capacité d'encours de travaux, de pièces, entre deux machines de production M_{i-1} et M_i implantées consécutivement sur le système flow-shop.
β_j^i	Date de début de réalisation du travail, de la pièce, J_j par la machine de production M_i .
ξ_j^i	Date de fin de réalisation du travail, de la pièce, J_j par la machine de production M_i .
τ	Délai moyen d'inter-fins de travaux, d'inter-sorties de pièces du système flow-shop. Cadence moyenne du système flow-shop.
τ_{max}	Délai moyen optimal d'inter-fins de travaux, d'inter-sorties de pièces du système flow-shop. Cadence moyenne optimale du système flow-shop.
φ_j^i	Échéance opératoire de la machine M_i pour le traitement de la pièce J_j .
$\varphi_j^i(t)$	Échéance opératoire de la machine M_i pour le traitement de la pièce J_j en cours de traitement à l'instant t .

$\phi_j^i(t)$	Laxité opératoire dont dispose la machine M_i à l'instant t avant l'échéance opératoire φ_j^i pour le traitement de la pièce J_j .
$\gamma_j^i(t)$	Temps opératoire résiduel à l'instant pour achever le traitement de la pièce J_j sur la machine M_i .
C_{max}	Makespan. Durée de réalisation complète d'un ensemble de travaux, de pièces, par le système flow-shop. Date de fin de réalisation du travail, de la pièce, J_n .
C_{min}	Durée minimale de réalisation complète d'un travail, d'une pièce, par le système flow-shop.
η_i	Séquence opératoire initiale affectée à la machine de production M_i .
\mathcal{S}_i	Ensemble des déclinaisons de la séquence opératoire initiale η_i affectée à la machine de production M_i .
\mathcal{C}_i	Nombre de séquences opératoires déclinées pour la machine de production M_i .
$\mathcal{C}_{\mathcal{M}}$	Nombre de combinaisons de la gamme opératoire exécutée par l'ensemble \mathcal{M} des machines de production du système flow-shop.

Table des figures

2.1	Système job-shop	24
2.2	Système flow-shop	25
2.3	Système flow-shop hybride	25
2.4	Hierarchie des types de systèmes de production	26
2.5	Enchaînement des productions	30
2.6	Positionnement des problèmes d'ordonnancement d'atelier	31
2.7	Diagramme temporel de réalisation de travaux	33
2.8	Relations entre les principaux critères réguliers	35
3.1	Modèle de base des systèmes étudiés	56
3.2	Système de file d'attente d'une phase de transfert	60
3.3	Modèle d'abstraction d'une phase de transfert	61
3.4	Diagramme d'exécution du premier exemple	67
3.5	Diagramme d'exécution du second exemple	69
3.6	Organisation hiérarchisée du système de pilotage	71
3.7	Exemple de chronogramme de réalisation d'un travail sur une machine	73
3.8	Modèle des systèmes contrôlés	74
4.1	Modélisation de la durée d'exécution d'une action	91
4.2	Modélisation d'une machine de production	92
4.3	Modélisation de l'interconnexion de deux machines	93
4.4	Modélisation de l'interaction non-préemptible du contrôleur avec une machine	95
4.5	Modélisation de l'interaction préemptible du contrôleur avec une machine	96
4.6	Modélisation de la concurrence d'accès au contrôleur	98

4.7	Modélisation haut niveau d'une machine de production	99
4.8	Modélisation haut niveau de l'interconnexion de deux machines	100
4.9	Modélisation haut niveau de l'interaction du contrôleur avec une machine	102
4.10	Modélisation haut niveau du mécanisme de préemption	104
4.11	Modélisation haut niveau simplifiée d'un système flow-shop contrôlé	105
5.1	Cadence de production pour un système à 2 machines	118
5.2	Cadence de production pour un système à 5 machines	118
5.3	Cadence de production pour un système à 8 machines	119
5.4	Cadence de production pour un système à 12 machines	119
5.5	Cadence de production pour un système à 20 machines	120
5.6	Cadence de production pour un système à 30 machines	120
5.7	Visualisation des temps de réponse du contrôleur	122
5.8	Simplification du modèle des systèmes contrôlés	124
5.9	Comparaison des bornes pour un système à 2 machines	126
5.10	Comparaison des bornes pour un système à 5 machines	126
5.11	Comparaison des bornes pour un système à 8 machines	127
5.12	Comparaison des bornes pour un système à 12 machines	127
5.13	Comparaison des bornes pour un système à 20 machines	128
5.14	Comparaison des bornes pour un système à 30 machines	128
5.15	Évolution des bornes mécanique et de contrôle	129
5.16	Diagramme d'exécution avec la règle de tri <i>FIFO</i>	133
5.17	Diagramme d'exécution avec la règle de tri <i>LIFO</i> non-préemptive	134
5.18	Diagramme d'exécution avec la règle de tri <i>LIFO</i> préemptive	135
5.19	Diagramme d'exécution avec la règle de tri <i>LPT</i> non-préemptive	136
5.20	Diagramme d'exécution avec la règle de tri <i>LPT</i> préemptive	137
5.21	Diagramme d'exécution avec la règle de tri <i>Pull</i> non-préemptive	138
5.22	Diagramme d'exécution avec la règle de tri <i>Pull</i> préemptive	139
5.23	Diagramme de démonstration de non-optimalité de la règle de tri <i>FIFO</i>	143
5.24	Diagramme d'exécution spécifique avec la règle de tri <i>FIFO</i>	145
5.25	Diagramme d'exécution spécifique avec la règle de tri <i>LPT</i> non-préemptive	146
6.1	Périodicité commune des machines de production	153
6.2	Synchronisation des machines de production	156

6.3	Visualisation du nombre de combinaisons pour une machine non-critique	161
6.4	Diagrammes d'exécution de machines synchronisées	164
7.1	Exemple d'exécution avec l'algorithme <i>EDF</i>	175
7.2	Exemple d'exécution avec l'algorithme <i>LLF</i>	175
7.3	Évaluation des échéances opératoires pour une machine	178
7.4	Résultat avec une borne mécanique et la règle de tri <i>EoDF</i>	181
7.5	Résultat avec une borne mécanique et la règle de tri <i>LoLF</i>	182
7.6	Résultat avec une borne de contrôle et la règle de tri <i>EoDF</i>	184
7.7	Résultat avec une borne de contrôle et la règle de tri <i>LoLF</i>	185
7.8	Résultat du cas non-optimal pour les deux règles de tri	187
D.1	Architecture de Munster	209

Liste des tableaux

2.1	Principaux critères d'optimisation	34
2.2	Valeurs usuelles du champ α	36
2.3	Extensions du champ α pour la spécification de flow-shop hybrides	37
2.4	Valeurs usuelles du champ β	38
2.5	Principales règles de tri	44
3.1	Données de paramétrage du premier exemple	66
3.2	Dates de début et de fin de travaux sur les machines du premier exemple	67
3.3	Données de paramétrage du second exemple	68
3.4	Dates de début et de fin de travaux sur les machines du second exemple	69
4.1	Comparaison des méthodes d'analyse de performances	86
4.2	Détail des transitions liées à l'interaction non-préemptible du contrôleur	101
4.3	Détail des transitions liées à l'interaction préemptible du contrôleur	103
4.4	Paramètres communs à l'ensemble du système	108
4.5	Délais d'assemblage pour chacune des machines	108
4.6	Évaluation de la cadence de production	109
5.1	Évaluation de la borne mécanique	116
5.2	Degré de productivité atteint par rapport à la borne mécanique	121
5.3	Évaluation de la borne de contrôle	124
5.4	Gamme opératoire pour les simulations complémentaires	132
5.5	Comparaison des cadences obtenues	140
5.6	Gamme opératoire pour un cas de non-optimalité de la règle de tri <i>FIFO</i>	142
6.1	Représentation d'une gamme opératoire	157

6.2	Évaluation d'une combinaison	163
6.3	Proposition de combinaisons optimales	164
6.4	Détail des combinaisons pour un cas sans solution optimale	167
6.5	Exemple d'une combinaison tolérante aux retards	169
7.1	Jeu de tâches évalué	174
7.2	Gamme pour l'évaluation des règles de tri avec une borne mécanique . . .	180
7.3	Gamme pour l'évaluation des règles de tri avec une borne de contrôle . . .	183
7.4	Gamme pour l'évaluation d'une configuration sans atteinte de l'optimalité	186
A.1	Complexités algorithmiques courantes	201