

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

No. attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

THESE

Pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : AUTOMATIQUE-PRODUCTIVE

préparée au Laboratoire d'Automatique de Grenoble

dans le cadre de l'**Ecole Doctorale**

Electronique, Electrotechnique, Automatique, Télécommunications, Signal

présentée et soutenue publiquement par

Jean-Marc BOLLON

le 11 mai 2001

Titre :

ETUDE DE DIFFERENTES POLITIQUES DE PILOTAGE DE SYSTEMES DE
PRODUCTION

Directeurs de thèse :

Maria DI MASCOLO (LAG)

Yannick FREIN (GILCO)

Jury :

Luc DUGARD (Président proposé)

Jean-Louis BOIMOND (Rapporteur)

Yves DALLERY (Rapporteur)

Maria DI MASCOLO (Directeur de thèse)

Yannick FREIN (Co-encadrant)

Fikri KARAESMEN (Examinateur)

Jean-Claude HENNET (Examinateur)

A Valéria
Et à toute ma famille

Remerciements

Je remercie les membres du jury pour l'honneur qu'ils me font de juger cette thèse :

Monsieur Luc DUGARD, Directeur de Recherche CNRS, Directeur du Laboratoire d'Automatique de Grenoble (LAG).

Monsieur Jean-Louis BOIMOND, Maître de Conférences à l'Université d'Angers ISTIA, Laboratoire d'Ingénierie des Systèmes Automatisés (LISA).

Monsieur Yves DALLERY, Professeur à l'Ecole Centrale de Paris, laboratoire Productique-Logistique (PL).

Madame Maria DI MASCOLO, Chargée de Recherche au CNRS, Laboratoire d'Automatique de Grenoble (LAG).

Yannick FREIN, Professeur à l'Ecole de Génie industriel de l'INPG, Directeur du laboratoire de Gestion Industrielle, Logistique et COncption (GILCO).

Fikri KARAESMEN, Chef de Travaux à l'Ecole Centrale de Paris, laboratoire Productique-Logistique (PL).

Jean-Claude HENNET, Directeur de Recherche au CNRS, Laboratoire d'Analyse et d'Architecture des Systèmes à Toulouse (LAAS-CNRS).

Je tiens avant tout à exprimer toute ma reconnaissance à Maria DI MASCOLO et Yannick FREIN qui, grâce à leur rigueur scientifique, leur disponibilité, leur expérience, et leur gentillesse m'ont permis d'améliorer la qualité de mes travaux, ainsi que leur avancement.

Je voudrais aussi remercier les permanents de l'équipe SYLODI (SYstèmes LOGiques et Discrets), René DAVID, Christian COMMAULT, Hassane ALLA, Mireille JACOMINO et bien sûr Maria DI MASCOLO, qui ont su faire partager, dans la convivialité, les travaux de chacun.

Je remercie aussi l'ensemble du personnel du LAG qui a toujours été très aimable, disponible et efficace.

Enfin je remercie Moez YEDDES, Tiberiu SAVA, Antonio FAVELA-CONTRERAS, Ahmad SKAF, et beaucoup d'autres doctorants et étudiants, avec qui j'ai pu créer des liens et partager des instants toujours très agréables et enrichissants.

Sommaire

Introduction	7
1.1 Les systèmes de production étudiés	8
1.2 Les objectifs de la thèse	9
1.3 Plan de la thèse	9
Description de politiques de gestion de production	11
2.1 Le basestock	13
2.2 Le kanban	15
2.3 Le kanban généralisé	16
2.3.1 Modélisation par réseau de files d'attente	17
2.3.2 Propriétés	17
2.4 Le kanban étendu	18
2.4.1 Modélisation	18
2.4.2 Propriétés	20
2.5 Autres politiques de gestion de production	20
2.5.1 Quelques variantes du Basestock	20
2.5.2 Quelques politiques avec production par lots	23
2.5.3 Quelques variantes du kanban	23
2.5.4 Politiques Finite-Buffer	24
2.5.5 Quelques politiques hybrides	24
2.5.6 Le MRP une politique à flux poussé	26
2.5.7 Le pilotage optimal	27
2.5.8 Quelques politiques pour des lignes avec assemblage	28
2.6 Conclusion	29
Eléments pour une formulation unifiée de la dynamique des politiques à flux tiré	31
3.1 Notations et éléments nécessaires à l'expression de notre formulation.	32

3.1.1	Description détaillée des mailles et des processus de fabrication	32
3.1.2	Description de variables dites "d'état global"	35
3.2	Dynamique et formulation des politiques pour une ligne avec stations monoserveur précédées d'un stock tampon.	36
3.2.1	Définition de la notion de dynamique	37
3.2.2	Définition de la notion de dynamique pour un critère de type (a)	37
3.2.3	Définition de la notion de dynamique pour un critère de type (b)	40
3.2.4	Détermination de la dynamique par F_i ou M_i en fonction de X	42
3.2.5	Conclusions pour une ligne avec stations monoserveur précédées de stocks tampon	53
3.3	Dynamique et formulation des politiques avec généralisation du type de processus de fabrication utilisé dans la ligne.	53
3.3.1	Définition de la notion de dynamique pour un critère de type (a)	54
3.3.2	Définition de la notion de dynamique pour un critère de type (b)	56
3.3.3	Détermination de la dynamique par F_i et C_i en fonction de X , ou par M_i et C_i en fonction de X	56
3.3.4	Conclusions pour des systèmes avec processus de fabrication quelconques à flux poussé	60
<i>Calcul des fonctions utilisées dans la formulation unifiée de la dynamique des politiques à flux tiré</i>		63
4.1	Rappels sur l'utilisation de l'algèbre des dioïdes appliquée aux graphes d'événements	64
4.2	Calcul de F_i en fonction de X pour quelques politiques à flux tiré.	67
4.2.1	Notations et éléments nécessaires au calcul de F_i en fonction de X .	67
4.2.2	Calcul de F_i en fonction de X pour le basestock	69
4.2.3	Calcul de F_i en fonction de X pour le kanban	70
4.2.4	Calcul de F_i en fonction de X pour le kanban étendu	71
4.2.5	Calcul de F_i en fonction de X pour le kanban généralisé	73
4.2.6	Résultats résumés pour les quatre politiques	75
4.3	Généralisation du calcul de F_i en fonction de X	76
4.4	Utilisation de F_i fonction de X pour différents processus de fabrication	80
4.4.1	Une station monoserveur avec stock tampon	81
4.4.2	Une station monoserveur sans stock tampon	81
4.4.3	Une station multiserveur avec stock tampon	82
4.4.4	Plusieurs stations monoserveur en série avec stock tampon	83
4.5	Application aux systèmes avec production par lots	84
4.5.1	Calcul de F_i pour une politique (Q, r^e) de type échelon	84

4.5.2	Calcul de F_i pour une politique (Q, r) de type installation	86
4.5.3	Calcul de F_i pour une politique Kanban avec production par lots	88
4.6	Application aux systèmes de production avec assemblage	90
4.6.1	Calcul de F_i pour un basestock avec assemblage	91
4.6.2	Calcul de F_i pour un kanban avec assemblage	92
4.7	Application au MRP	94
4.8	Conclusions	95

Recherche des identités de fonctionnement entre politiques _____ **97**

5.1	Quelques équivalences de comportement connues	98
5.2	Pour une ligne de deux étages, comparaison entre kanban étendu et kanban généralisé	99
5.2.1	Processus de fabrication quelconque précédé d'un stock tampon et tel que les pièces sortent une à une.	100
5.2.2	Processus de fabrication avec monoserveur précédé d'un stock tampon.	102
5.3	Kanban étendu et kanban généralisé sur une ligne de plusieurs étages	103
5.3.1	Comparaisons avec paramétrage à valeur finie	103
5.3.2	Comparaisons avec des paramètres K_i de valeur infinie.	113
5.4	Conclusion sur la comparaison des politiques	120

Comparaison de coûts entre des politiques sous-optimales avec pour référence la politique optimale. _____ **123**

6.1	Introduction	123
6.2	Calcul de politiques optimales par programmation dynamique.	125
6.2.1	Politique optimale pour un coût pondéré	128
6.2.2	Politique optimale pour un coût moyen	131
6.3	Caractérisation partielle des politiques.	133
6.4	Comparaisons entre des politiques sous-optimales et la politique optimale pour une ligne de deux étages.	136
6.5	Comparaisons pour une ligne de deux mailles avec délai	140
6.5.1	Cas d'un livraison au plus tôt avec délai	141
6.5.2	Cas du système avec livraison après délai	146

6.6	Comparaisons pour un système de deux étages avec production par lots et délai avec livraison au plus tôt	152
6.6.1	Description	152
6.6.2	Politique optimale	153
6.6.3	Comparaisons	155
6.7	Système de deux étages avec assemblage	156
6.7.1	Description	156
6.7.2	Politique optimale	157
6.7.3	Politique kanban	158
6.8	Conclusion	159

Quelques tentatives de développement d'une méthode de calcul des coûts moyens commune à toutes les politiques. _____ **161**

7.1	Utilisation de la méthode de la matrice géométrique de Neuts	162
7.1.1	Numérotation des états	162
7.1.2	Méthode de calcul	165
7.1.3	Résultats numériques	167
7.1.4	Calcul du coût moyen et de quantités moyennes	170
7.2	Autre méthode pouvant donner des solutions exactes en nombres rationnels	171
7.2.1	Méthode de numérotation des états	171
7.2.2	Méthode de calcul des probabilités	176
7.2.3	Résultats numériques pour « des cas simples »	178
7.2.4	Calcul du coût moyen et de quantités moyennes	181
7.2.5	Résolution de cas plus complexes	182
7.3	Conclusion sur ces méthodes de résolution de la chaîne de Markov	184

Conclusion _____ **187**

8.1	Le bilan	187
8.2	Perspectives	190

Bibliographie _____ **193**

<i>Annexe A : Notations</i>	199
 <i>Annexe B : Eléments annexes du chapitre 3</i>	 201
B.I Preuve d'un lemme et d'une proposition de la section 3.2.4.2 : _____	201
B.II Dynamique pour un critère de type (a) généralisant les types de processus de fabrication utilisables. _____	202
B.III Dynamique pour un critère de type (b) généralisant les types de processus de fabrication utilisables. _____	204
B.IV Preuves et éléments de la section 3.3.3.1 : _____	205
B.V Section 3.3.3.2 : Détermination de la dynamique par M_i et C'_i en fonction de X pour un critère de type (b) . _____	208
B.VI Section 3.3.3.3 : Relations entre les deux types de dynamique pour des processus de fabrication avec stock tampon et multiserveur. _____	210

Chapitre 1

Introduction

Depuis de nombreuses années l'industrie cherche à réduire ses coûts de production en améliorant la gestion de ses systèmes de production. La limitation de la taille des stocks est un des principaux enjeux de cette gestion. Cependant ces stocks sont indispensables à la bonne satisfaction des demandes. Lorsque celles-ci ne sont pas servies à temps, il s'ensuit souvent une coûteuse perte de clientèle. Il faut donc parvenir à trouver un compromis entre ces deux objectifs antagoniques.

Pour atteindre ce compromis, de nombreuses méthodes ou politiques de gestion de production ont été développées. Elles se composent d'un certain nombre de règles permettant de commander le fonctionnement du système de production. La problématique consiste alors à choisir la bonne politique.

Un premier élément de comparaison des politiques de gestion de production consiste à regarder si elles peuvent avoir un comportement identique. Parfois la façon de décrire les politiques masque des similitudes de fonctionnement. Cependant, même lorsque avec un même formalisme, les descriptions sont différentes, il n'est pas exclu qu'il y ait des identités de fonctionnement.

Un autre élément de comparaison consiste à calculer les coûts (de possession des produits et la non satisfaction de la demande) engendrés par un système de production en fonction de la politique choisie. Cela permet d'établir une classification entre chaque politique, dépendant souvent des données initiales du problème. Cependant il est parfois possible de montrer qu'une politique sera toujours moins coûteuse qu'une autre. Dans certains cas, il est même possible de calculer la politique qui a le plus petit coût. Elle est alors qualifiée d'optimale tandis que les autres sont sous-optimales. La plupart du temps, ces politiques optimales ne peuvent être mises en œuvre à cause de leur grande complexité.

Le calcul des coûts engendrés par une politique est souvent difficile à obtenir et suivant les méthodes, les temps de calculs peuvent être importants. Les méthodes de simulation, très souples à l'usage, sont souvent employées. Cependant celles-ci ne sont pas assez rapides car une politique possède presque toujours des paramètres qu'il faut optimiser, ce qui nécessite le calcul de nombreuses configurations. On peut aussi utiliser des méthodes analytiques qui sont

plus efficaces. Malheureusement, presque chaque politique nécessite une technique de calcul qui lui est propre. De plus, pour certaines politiques, il n'y a pas de méthode analytique disponible, ou bien il peut être nécessaire de faire des hypothèses restrictives.

1.1 Les systèmes de production étudiés

Les systèmes de production que nous considérons sont constitués de mailles que l'on appelle aussi étages (figure 1.1). Chaque maille contient un processus de fabrication (mécanisme de transformation des produits pouvant être constitué de plusieurs machines) suivi d'un stock de produits intermédiaires semi finis, ou de produits finis si la maille se situe complètement en aval. La maille la plus en amont est supposée être précédée d'un stock infini de produits bruts.

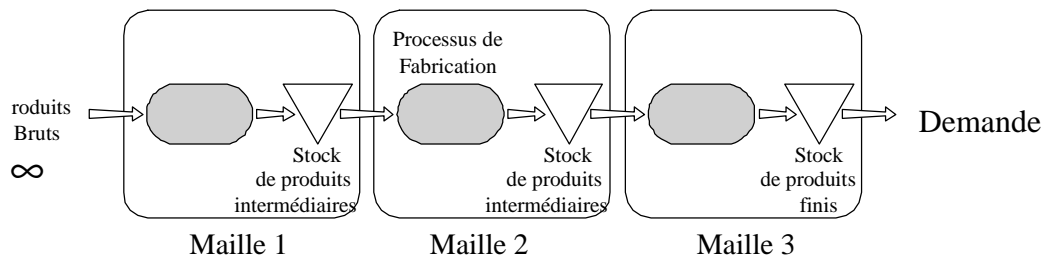


Figure 1.1 : Ligne de production composée de trois mailles (ou étages)

Nous avons fait l'hypothèse que les processus de fabrication produisent un seul type de produit et que les temps de production sont stochastiques. De façon marginale, nous avons aussi traité le cas d'une production par lots. La demande est généralement supposée poissonnienne. Nous avons aussi étudié le cas d'un système où la demande peut ne pas être satisfaite avant un certain délai.

Nous ne nous sommes pas intéressés aux politiques fonctionnant à flux poussé mais seulement à celles fonctionnant à flux tiré. On parle de flux poussé lorsque la gestion de production s'effectue en poussant les produits dans les processus de fabrication pour satisfaire la prévision de la demande. Les politiques à flux tiré quant à elles ont pour principe de réagir à l'arrivée des demandes fermes. Des produits sont tirés en aval par la demande. Pour éviter qu'un nombre excessif de produits soient en circulation dans la ligne, le flux de pièces peut être interrompu temporairement à l'entrée des machines. On peut ainsi éviter d'atteindre certains états indésirables. Une production à flux tiré peut se faire pour maintenir les stocks à un certain niveau (production pour stock) ou uniquement pour satisfaire la demande en évitant tout stockage (production pour ordre). Dans le cas d'une production pour stock, lorsque le système est au repos et qu'il n'y a pas de demandes, les stocks ont un certain niveau et ils ne sont pas vides ; par contre dans le cas d'une production pour ordre ils doivent tous être vides.

La structure des systèmes de production que nous avons étudiés est principalement celle d'une ligne de production, c'est-à-dire d'un enchaînement de mailles en série. Quelques cas de système de production avec assemblage ont aussi été considérés.

1.2 Les objectifs de la thèse

Notre objectif initial a été la comparaison et l'étude de politiques optimales et sous-optimales, pour la minimisation d'une fonction coût. Les politiques optimales ont été obtenues par un calcul de pilotage optimal effectué par programmation dynamique. Les politiques usuelles sous-optimales ont pu elles aussi être décrites sous forme de pilotage. Nous nous sommes alors aperçus qu'il était possible de décrire un très grand nombre de politiques, à l'aide d'une formulation mathématique assez simple, n'utilisant que quelques constantes de paramétrage. Nous avons alors étendu cette formulation à différentes catégories de systèmes de production.

La simplicité de cette formulation nous a permis de mettre au point une technique de recherche des identités de fonctionnement entre politiques.

Elle a aussi permis l'élaboration de méthodes de calcul exactes des probabilités de chaque état d'un système de production. Cette méthode peut s'appliquer à un très grand nombre de politiques et les résultats numériques sont parfois obtenus sous formes de fractions rationnelles exactes.

1.3 Plan de la thèse

Dans le chapitre deux une revue bibliographique des politiques est faite. Nous décrivons principalement les politiques basestock, kanban, kanban généralisé et kanban étendu. Quelques politiques utilisées pour des systèmes de production avec assemblage ou production par lots sont aussi exposées.

Le chapitre trois introduit notre méthode de formulation des politiques. Il s'agit surtout de démontrer que cette formulation est suffisante à la description d'un système de production. Cette formulation a été initialement utilisée pour de petits problèmes de pilotage optimal où le processus de fabrication est représenté par une station exponentielle. Nous avons alors démontré qu'il était possible d'utiliser cette formulation avec des processus de fabrication pratiquement quelconques. Quelques limites théoriques à l'usage de cette formulation ont pu être définies (limites dépendant de la politique ou des processus de fabrication).

On poursuit ensuite dans le chapitre quatre par le calcul des fonctions déterminant la formulation. Ce calcul est réalisé pour différentes politiques (basestock, kanban, kanban

étendu ...) et différents systèmes de production (simples lignes de production, systèmes produisant par lots ou possédant des stations d'assemblage). Pour trouver efficacement les fonctions permettant de formuler certaines politiques, quelques notions de l'algèbre des dioïdes sont utilisées (l'algèbre $(\min, +)$ en particulier). Il y est alors démontré qu'une formulation que nous avons qualifiée de canonique, permet de modéliser la totalité d'une classe très générale de politiques à flux tiré.

Dans le chapitre cinq, nous recherchons, à partir de cette formulation dite canonique, toutes les identités de fonctionnement qui peuvent exister entre deux politiques telles que le kanban généralisé et le kanban étendu.

Dans le chapitre six, nous calculons la politique de pilotage optimale de plusieurs systèmes de production. L'objectif est de trouver le moins coûteux des pilotages contrôlant la marche des machines. Ce pilotage est fonction de l'état du système, il est donné par des courbes de commande. Chaque courbe définit la commande d'une machine en séparant les états de marche des états d'arrêt de la machine. Le coût de ces politiques optimales sert alors de référence à la comparaison de politiques sous-optimales.

Comme pour la politique optimale, les politiques sous-optimales peuvent aussi être exprimées par des courbes de commande. Les courbes de commande sont d'ailleurs à l'origine de notre formulation des politiques. Pour effectuer le calcul du coût des politiques sous-optimales nous réutilisons, grâce aux courbes de commande et avec quelques modifications, la technique de calcul qui a servi à trouver le coût des politiques optimales.

Plusieurs systèmes de production sont étudiés, mais celui qui a le plus retenu notre attention est une ligne de deux étages où l'on considère l'existence d'un délai de livraison. On distingue deux cas, un qui consiste à ne pas livrer avant le délai et un autre où l'on peut livrer dès que possible. Nous avons aussi traité le cas de systèmes produisant par lots et celui d'un système composé de deux mailles suivies en aval d'une maille d'assemblage.

Dans le dernier chapitre deux méthodes de calcul de la probabilité des états et des coûts sont proposées pour des lignes de production à deux mailles. Ces méthodes ne s'appliquent que si dans la politique utilisée le nombre de produits est borné au niveau du stock de produits intermédiaires et de l'encours de la station livrant des produits finis. Lorsque cette borne est petite il a même été possible, avec une des méthodes, de trouver des solutions en nombres rationnels, c'est-à-dire écrites sous forme de fraction.

Enfin quelques conclusions et perspectives sont données.

Chapitre 2

Description de politiques de gestion de production

Une politique de gestion de production est un ensemble de règles qui permettent de commander ou de piloter les différents processus de fabrication d'un système. Ces politiques doivent permettre d'utiliser au mieux un système de production donné. Pour choisir la bonne politique il faut commencer par fixer des critères. Selon les critères et le système de production étudié il est parfois possible de trouver la meilleure des politiques, elle est alors nommée la politique optimale. Souvent une politique optimale sera trop complexe et sans intérêt pratique car impossible ou trop coûteuse à mettre en œuvre. Par contre certaines politiques non optimales, que l'on nomme politiques sous-optimales, peuvent être assez simples à mettre en œuvre. Pour cette raison, et comme la plupart du temps on ne sait pas trouver la politique optimale, ces politiques sous-optimales sont très utilisées.

Hormis l'aspect optimal ou non optimal, il existe deux grandes classes de politiques de gestion de production, ce sont les politiques dites à flux poussé et celles à flux tiré. De façon un peu caricaturale, on peut dire que les premières vont pousser, en fonction des demandes prévisionnelles, des produits bruts dans le système de production et que les secondes tirent le flux de production, en fonction des demandes réelles qui arrivent à chaque instant. Ces deux types de politiques utilisent différemment l'information pour gérer la production. Les politiques à flux poussé ont besoin d'une prévision fiable de la demande tandis que celles à flux tiré n'utilisent pas de prévisions mais seulement les demandes fermes. Cependant la frontière entre politique à flux tiré et politique à flux poussé n'est pas toujours aussi claire. Par exemple en temporisant l'arrivée des demandes prévisionnelles, il peut être possible de voir un système de production à flux poussé comme un système de production à flux tiré [KAR 1999]. L'objet de nos recherches portera sur les systèmes de production à flux tiré.

Pour la plupart des politiques de gestion de production à flux tiré, on utilise une méthode de production pour stock. Dans ces méthodes de production, on définit des niveaux de stock de produits finis ou semi finis, il faut alors produire pour maintenir ces niveaux puisque la demande provoque leur diminution. Chaque fois qu'une demande arrive dans le système, le stock de produits finis diminue ou parfois le nombre de demandes non servies augmente.

Les politiques à flux tiré doivent gérer les flux de production de telle sorte qu'il n'y ait ni trop de pièces dans le système ni pas assez. Ainsi, l'objectif est de contrôler le nombre de pièces présentes dans les stocks et processus de fabrication afin de trouver un compromis entre les coûts de possession des produits et les coûts ou contraintes de satisfaction de la demande. Pour effectuer ce contrôle les politiques à flux tiré réagissent à la demande client. Cette demande est envoyée au niveau de chacune des mailles du système de production pour permettre de rétablir le niveau du stock de produits finis et de chacun des stocks de produits intermédiaires. Cependant il est parfois nécessaire de ralentir la remontée de cette information pour éviter, par exemple, d'avoir une accumulation de pièces devant une machine qui n'arrive pas à suivre la cadence. Par contre si la demande n'est pas remontée suffisamment rapidement, le système risque de prendre du retard pour le service des demandes.

Dans un système géré à flux tiré, c'est la demande qui tire la production. Cependant à l'intérieur de chaque maille, la production dans les processus de fabrication, s'effectue à flux poussé car la circulation des pièces ne dépend que des instants où elles sont libérées dans ces processus de fabrication. Il n'est pas possible d'intervenir sur une pièce qui a été introduite dans un processus de fabrication et rien ne peut entraver la sortie de celle-ci en direction du stock de produits intermédiaires ou de produits finis qui suit. Cependant, les pièces peuvent parfois être bloquées à la sortie du processus de fabrication par un contrôle en fin de service ([BON 1995], [BER 1991]), ce dernier cas ne sera pas considéré.

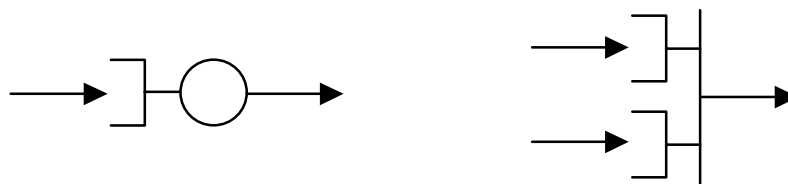


Figure 2.1 : Une file d'attente suivie d'un serveur et à droite deux files d'attente suivies d'une station de synchronisation.

La description des politiques prend des formes très diverses et il existe parfois plusieurs façons de décrire une même politique. Nous utiliserons fréquemment des réseaux de files d'attente. Ceux-ci sont constitués par des serveurs, des files d'attente et des stations de synchronisation, qui sont reliés par des arcs (voir figure 2.1). Les éléments qui circulent dans ces réseaux sont appelés clients (pièces, demandes, ordres de fabrication,...). Les files d'attente permettent de faire attendre des clients avant qu'ils ne puissent changer de localisation dans le réseau. En plus du formalisme très classique des réseaux de files d'attente [BAY 2000], nous utilisons beaucoup les stations de synchronisations [LIB 2000]. Lorsqu'une station de synchronisation est franchie, il faut pouvoir retirer un client de chaque file amont et envoyer un client sur chaque arc sortant. Pour noter le nombre de clients présents dans une station, ou une file d'attente, on utilisera leur nom écrit en italique. Le

nombre de clients dans une file d'attente Q_i sera donc Q_i . Voici à présent une description de quelques politiques rencontrées dans la littérature.

2.1 Le basestock

Cette politique consiste à maintenir chaque stock P_i à un niveau nominal propre S_i qui est supposé atteint à l'instant initial. Afin d'aboutir cet objectif, une autorisation de production pour une pièce est envoyée à tous les étages dès qu'une demande de produits finis arrive. Ainsi la production ne s'arrête que lorsque tous les stocks ont atteint leur niveau nominal.

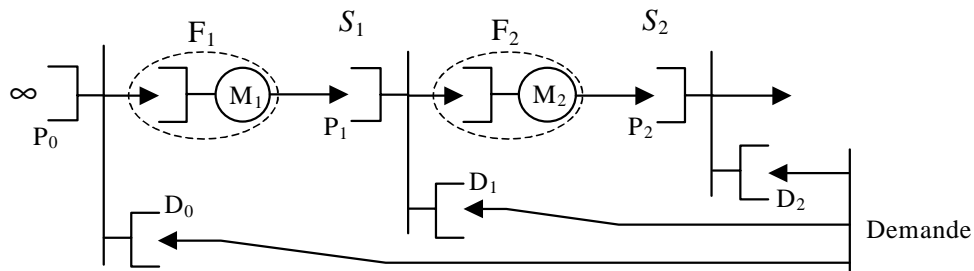


Figure 2.2 : Basestock pour une ligne de deux étages (ou mailles)

On peut représenter cette politique par un réseau de files d'attente ; la figure 2.2 montre un exemple de basestock pour une ligne de deux mailles ($N = 2$). Pour chaque étage i , le processus de fabrication F_i (en pointillé sur la figure) est ici représenté par une file d'attente suivie d'un serveur. Cette file d'attente a son importance car même si le serveur qui suit est saturé, le franchissement de la station de synchronisation qui précède, sur laquelle se trouve le stock de produits intermédiaires P_{i-1} , reste alors possible.

Quand une demande arrive, elle est transmise immédiatement vers les files D_N , D_i et D_0 qui se synchronisent respectivement avec le stock de produits finis P_N , les stocks de produits intermédiaires P_i et le stock de produits bruts P_0 qui est infini. S'ils ne sont pas vides, chacun de ces stocks se vide d'une pièce qui est envoyée dans le processus de fabrication suivant ou vers le client quand il s'agit d'un produit fini. Lorsqu'un des stocks P_i est vide, aucune pièce ne peut être envoyée dans la station qui suit. Les demandes transmises sont alors gardées dans une file d'attente D_i et représentent une accumulation d'autorisations de production ou une accumulation de demandes non servies lorsque $i = N$.

Le blocage du passage des clients sur une station de synchronisation ne peut être causé ici que par la présence de files d'attente vides. Si le passage des clients est possible alors les files se vident, ainsi pour chaque étage i , au moins une des files D_i ou P_i est vide. En notant le nombre d'éléments contenus dans les files d'attente D_i et P_i respectivement par D_i et P_i , on obtient l'équation (2.1) suivante qui est valable à chaque instant :

$$\min(D_i, P_i) = 0 \quad \text{pour } i = 1 \cdots N \quad (2.1)$$

Comme le stock de produits bruts n'est jamais vide alors $D_0 = 0$. Pour comprendre une politique ou pour faire des calculs, on utilise souvent des invariants. Ce sont des constantes obtenues en additionnant ou en retranchant un certain nombre d'éléments. Pour le basestock on a les invariants suivants [DUR 1997] :

$$D_{i-1} + F_i + P_i - D_i = S_i \quad \forall i = 1 \dots N \quad (2.2)$$

où F_i est le nombre de pièces présentes dans la file F_i . On peut ainsi observer qu'il n'y a pas de limite au nombre de pièces présentes dans la ligne si le système n'arrive pas à suivre la demande. En effet le nombre de pièces présentes dans une ligne est $\sum_{i=1}^N (P_i + F_i) = D_N + \sum_{i=1}^N S_i$, soit le nombre de demandes en attente, qui n'est pas borné, plus la somme des niveaux nominaux. Cette non limitation du nombre de pièces présentes dans la ligne est le principal inconvénient du basestock. Par contre cette politique a l'avantage d'être très réactive à la demande car chaque demande se transforme instantanément en un ordre de fabrication au niveau de chacun des processus de fabrication.

En général cette politique n'est pas optimale. Veatch et Wein [VEA 1994] ont montré qu'elle ne l'était jamais pour le cas d'un système de deux mailles composées de stations monoserveurs exponentielles, avec une demande poissonnienne et où le critère est un coût linéaire de possession et de non satisfaction de la demande. Inversement pour un système où les stations sont des serveurs infinis [SCA 1960], [CLA 1960], ou lorsqu'il n'y a qu'une maille [FED.I 1986] et [FED.II 1986], il a été montré que le basestock pouvait être optimal.

La politique basestock est parfois décrite par un mécanisme de maintien de l'inventaire contenu dans certaines zones. Dans [BON 1996] ces zones d'inventaire sont représentées par des cellules de contrôle.

La figure 2.3 illustre une telle représentation pour une ligne composée de deux mailles. Les ovales représentent ici les processus de fabrication qui sont composés d'un stock tampon et d'un serveur. Lorsque le processus de fabrication ne possède pas de stock tampon il sera décrit par un cercle qui représente le serveur.

A l'intérieur des cellules de contrôle il faut pousser des pièces sans que jamais le nombre de pièces présentes dans une cellule moins le nombre de demandes en attente ne dépasse un certain niveau. Ces cellules contiennent un processus de fabrication et les mailles qui suivent plus en aval. Le niveau à ne pas dépasser est égal à la somme des S_i de chaque maille i recouverte par la cellule.

Dans la littérature, la politique basestock est une des politiques de gestion de production les plus utilisées car elle est relativement ancienne (décrite par Kimball durant les années 1950 mais publié plus tard dans [KIM 1988]), simple et facile à mettre en œuvre. Plusieurs variantes du basestock existent et sont décrites dans la section 2.5.1. On peut noter que si on se limite à un système de production où les pièces sont servies une à une, ces variantes sont la plupart du temps équivalentes au basestock de base décrit dans cette section.

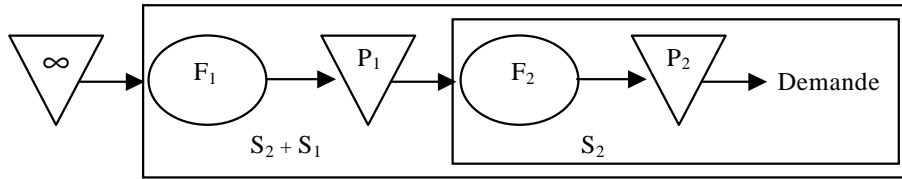


Figure 2.3 : Cellules de contrôle du basestock pour une ligne de deux mailles

2.2 Le kanban

Le kanban est une politique de gestion de production qui utilise pour son fonctionnement des étiquettes. Le mot kanban signifie étiquette en japonais. Cette politique provient donc du Japon et une de ses premières utilisations a été faite dans les usines de Toyota dès 1962, [SUG 1977]. Son principe consiste à maintenir un nombre limité de pièces dans chacune des mailles du système de production. Pour atteindre cet objectif, on attribue K_i étiquettes à chaque maille i . Pour qu'une pièce puisse être présente dans une maille, il faut lui attacher une étiquette. Ainsi, dès qu'une étiquette est disponible dans une maille une pièce peut entrer. Lorsqu'une pièce sort d'une maille, pour satisfaire une demande ou pour entrer dans une nouvelle maille, son étiquette est détachée et permet ainsi l'entrée d'une nouvelle pièce en remplacement. Le nombre de pièces présentes dans une maille i est donc limité à K_i . On remarque aussi que seule une sortie de pièce d'un étage permet une remontée de la demande vers l'amont. Donc la demande ne peut pas remonter en amont d'un stock vide.

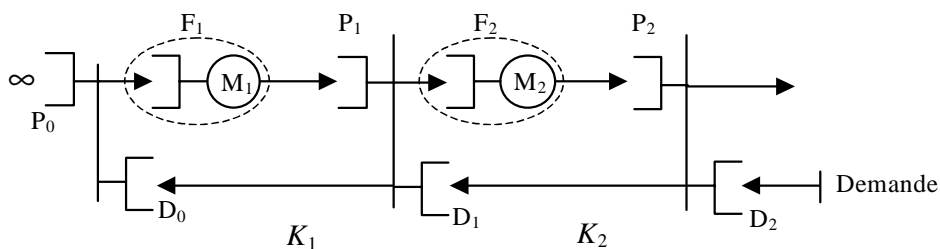


Figure 2.4 : Kanban pour une ligne de deux étages

La politique kanban peut être représentée par un réseau de files d'attente. La figure 2.4 en montre un exemple pour une ligne de deux étages ($N = 2$). Le processus de fabrication F_i d'une maille i est représenté par une file d'attente tampon suivie d'un serveur. La file d'attente D_N accumule les demandes non servies tandis que les files D_i pour $i \neq N$ contiennent les étiquettes disponibles qui attendent l'arrivée d'une pièce dans le stock P_i . Puisque P_0 n'est jamais vide, D_0 sera toujours nul. Dans ce réseau de files d'attente, on a la propriété qui suit,

$$\min(D_i, P_i) = 0 \quad \text{pour } i = 1 \dots N \quad (2.3)$$

et les invariants suivants :

$$D_{i-1} + F_i + P_i = K_i \quad \forall i = 1 \dots N \quad (2.4)$$

Il est aussi possible de décrire un système kanban à l'aide de cellules de contrôle (figure 2.5). Chaque maille i est alors contenue dans une cellule qui ne doit pas renfermer plus de K_i pièces. La politique consiste alors à pousser des pièces dans les processus de fabrication tout en contrôlant si la capacité des cellules n'est pas dépassée.

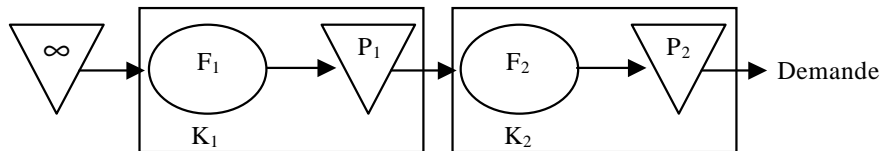


Figure 2.5 : Cellules de contrôle du kanban pour une ligne de deux mailles

Le kanban a l'avantage de pouvoir garantir quel sera le nombre maximal de pièces présentes dans chaque maille. Cependant, contrairement au basestock, la demande qui est transmise en amont par un système kanban est arrêté provisoirement dès qu'un stock de produits vide est rencontré. Ce système est donc moins réactif vis-à-vis de la demande que le basestock, mais il n'en a pas les inconvénients car les encours sont limités par le nombre d'étiquettes.

2.3 Le kanban généralisé

Cette politique qui a été principalement mise au point par Buzacott [BUZ 1989.a] est issue du système P.A.C ("Production Autorisation Card") [BUZ 1993]. Ce système P.A.C. est capable de modéliser plusieurs politiques de gestion de production comme le basestock le kanban et aussi le MRP qui est une politique à flux poussé. En simplifiant ce système P.A.C, on obtient la politique kanban généralisé, considérée comme une politique hybride du basestock et du kanban car elle peut modéliser l'une ou l'autre. Cette politique utilise deux paramètres par maille (S_i et K_i) qui permettent de faire un compromis entre les qualités et les défauts du kanban et du basestock. Des études quantitatives [DUR 1997] ont d'ailleurs permis de montrer l'intérêt de cette politique vis à vis des politiques kanban et basestock.

Dans son fonctionnement le kanban généralisé va maintenir un niveau nominal S_i dans chaque stock P_i d'une maille i . Comme pour le kanban et le basestock les processus de fabrication ne doivent pas générer de blocages du flux de production, ils sont donc composés d'une file d'attente suivie d'un serveur. Cependant le kanban généralisé limite le nombre de pièces présentes dans chaque processus de fabrication F_i par une boucle de K_i étiquettes kanban. Ces étiquettes autorisent l'entrée des pièces dans les processus de fabrication puis elles sont ensuite détachées en sortie de ceux-ci quand la pièce entre dans le stock qui suit. Une fois libérées elles ne peuvent remonter pour autoriser la fabrication que s'il y a des

demandes à transmettre en amont. Réciproquement, pour que l'information associée à la demande puisse remonter, une étiquette libre doit être présente. Cette remontée effectuée, les étiquettes vont autoriser l'entrée d'une nouvelle pièce dans le processus de fabrication de la maille tandis que la demande continue à monter en direction des mailles situées plus en amont. Le rôle de ces étiquettes est donc double. Elles permettent de limiter le nombre de pièces circulant dans le processus de fabrication mais elles vont aussi freiner le flux des demandes qui remontent vers les étages amont.

2.3.1 Modélisation par réseau de files d'attente

La figure 2.6 illustre par un réseau de files d'attente le fonctionnement du kanban généralisé sur une ligne de deux étages. Pour un étage i , les K_i étiquettes du kanban généralisé ne sortent pas du circuit constitué du processus de fabrication F_i et des files B_i et A_{i-1} . La demande remonte progressivement pour permettre la transmission des ordres de fabrication nécessaires au maintien de chaque stock P_i à un niveau S_i . Cette remontée des demandes peut être interrompue dès qu'une file B_i ne contenant plus d'étiquettes est rencontrée. Pour débloquer la situation une étiquette doit être libérée. Il faut donc attendre qu'une pièce sorte du processus de fabrication F_i . Les files d'attente A_i pour $i < N$ contiennent des étiquettes qui ont été associées à une demande ; elles deviennent alors des autorisations de production et chacune va pousser une pièce dans le processus de fabrication F_i qui suit.

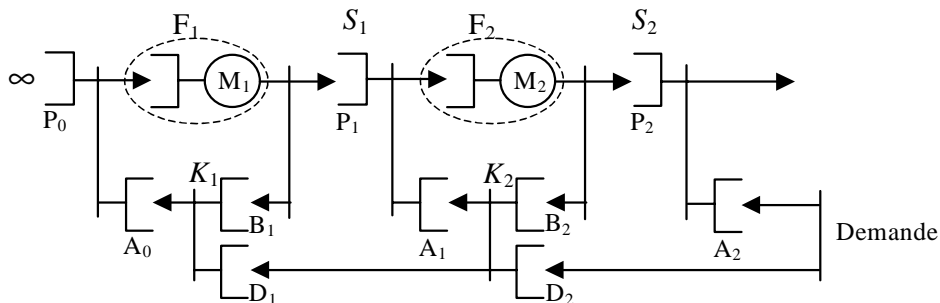


Figure 2.6 : Kanban généralisé pour une ligne de deux mailles

2.3.2 Propriétés

En partant du réseau de files d'attente de la figure 2.6 et en supposant qu'à l'état initial seules les files B_i et P_i ne sont pas vides avec des niveaux respectifs K_i et S_i , alors il est possible de trouver deux types d'invariants (voir [DUR 1997]). Le premier est proche de celui trouvé pour le basestock et le second proche de celui trouvé pour le kanban :

$$D_i + A_{i-1} + F_i + P_i - A_i = S_i \quad \forall i = 1 \dots N \quad (2.5)$$

$$A_{i-1} + F_i + B_i = K_i \quad \forall i = 1 \dots N \quad (2.6)$$

Pour chaque station de synchronisation il n'y a qu'une file d'attente non vide :

$$\min(D_i, B_i) = 0 \quad \text{pour } i = 1 \dots N \quad (2.7)$$

$$\min(A_i, P_i) = 0 \quad \text{pour } i = 0 \dots N \quad (2.8)$$

On peut aussi à partir de ces résultats trouver les bornes qui suivent [FRE 1995]

$$F_i \leq K_i \quad \text{pour } i = 1 \dots N \quad (2.9)$$

$$P_i \leq S_i \quad \text{pour } i = 1 \dots N \quad (2.10)$$

$$F_i + P_i \leq \max(K_i, S_i) \quad \text{pour } i = 1 \dots N \quad (2.11)$$

$$B_i \geq K_i - S_i - K_{i+1} \quad \text{pour } i = 1 \dots N - 1 \quad (2.12)$$

Le nombre de pièces présentes dans les stocks P_i et les processus de fabrication F_i est donc limité. La file d'attente B_i contient au minimum $K_i - S_i - K_{i+1}$ étiquettes. Si ce nombre est strictement positif il y aura des étiquettes inutiles. Pour ne pas en avoir, il suffit d'avoir la condition nécessaire suivante [BUZ 1989.a] :

$$K_i \leq S_i + K_{i+1} \quad \text{pour } i = 1 \dots N - 1 \quad (2.13)$$

Lorsque le nombre d'étiquettes kanban est infini dans toutes les mailles, la remontée des demandes est instantanée et la politique devient identique à celle du basestock. Le kanban généralisé peut aussi être identique à la politique kanban à condition que, pour tout i , on ait K_i égal à S_i [DUR 1997].

2.4 Le kanban étendu

Le kanban étendu est, comme le kanban généralisé, une politique hybride du basestock et du kanban. Elle a été développée plus récemment par Dallery et Liberopoulos [DAL 1995] et [DAL 2000].

Comme pour le basestock, les demandes remontent immédiatement à chaque étage pour autoriser la production d'une pièce, cependant le nombre de pièces par étage est limité par la présence d'un nombre fini de kanbans. Dans une maille i , le niveau de stock S_i qu'il faut atteindre ne peut être supérieur à K_i , le nombre de kanbans, puisque ceux-ci limitent la quantité de pièces présentes. Par conséquent le paramétrage de cette politique doit vérifier $K_i \geq S_i$.

2.4.1 Modélisation

Un réseau de files d'attente permet de décrire facilement ce mécanisme. Une ligne de deux étages de type kanban étendu est illustrée par la figure 2.7.

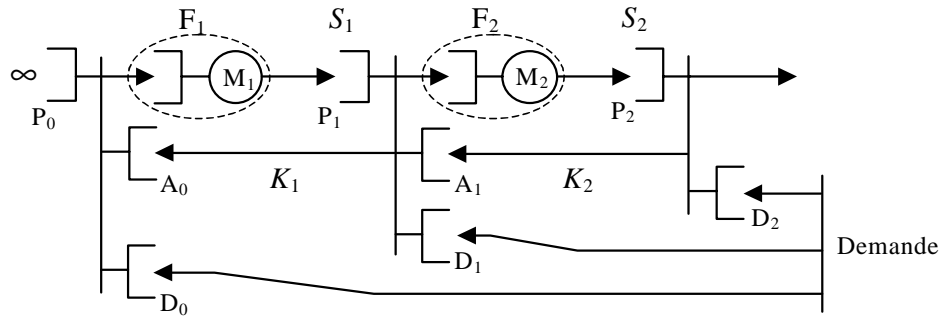


Figure 2.7 : Kanban étendu pour une ligne de deux étages

Le fonctionnement détaillé du réseau de files d'attente utilisé pour cette politique est le suivant :

Pour un étage i , les pièces sortent du processus de fabrication F_i pour aller dans le stock P_i . Chaque demande de produit fini génère un ordre de fabrication qui va dans les files D_i pour i allant de 0 à $N-1$ et celui-ci n'est pris en compte que s'il y a des pièces et des étiquettes kanban disponibles dans les files d'attente P_i et A_i . La file d'attente D_N conserve les demandes client non servies.

Ce système peut aussi être modélisé à l'aide de cellules de contrôle. Il suffit alors de superposer des cellules de contrôle de type basestock à des cellules de type kanban. La figure 2.8 montre un exemple de kanban étendu modélisé de la sorte.

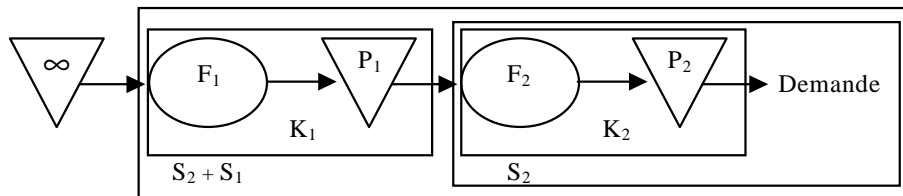


Figure 2.8 : Cellules de contrôle d'un kanban étendu pour une ligne de deux mailles

Chaque cellule de contrôle exprime une contrainte qui va limiter les flux de pièces entrant dans les processus de fabrication. Les deux cellules de la figure 2.8 qui « entourent » la demande, correspondent aux contraintes suivantes :

$$F_2 + P_2 - D_2 \leq S_2$$

$$F_1 + P_1 + F_2 + P_2 - D_2 \leq S_1 + S_2$$

les deux autres cellules qui « n'entourent pas » la demande représentent les contraintes

$$F_2 + P_2 \leq K_2$$

$$F_1 + P_1 \leq K_1$$

Ces quatre contraintes suffisent à définir la politique kanban étendu pour une ligne de deux étages.

2.4.2 Propriétés

A l'état initial on suppose que pour chaque maille i seules les files P_i et A_{i-1} ne sont pas vides avec des niveaux respectifs S_i et $(K_i - S_i)$.

Le franchissement des stations de synchronisation en début de maille ne peut pas dépendre des processus de fabrication qui suivent car leur file d'en-cours empêche la création de blocages. Le passage sur ces stations de synchronisation ne dépend que du niveau des files d'attente situées en entrée. Ainsi en amont de chaque étage i au moins une des files P_{i-1} , D_{i-1} ou A_i est vide. On obtient donc l'équation qui suit.

$$\min(D_i, A_i, P_i) = 0 \quad \text{pour } i = 0 \dots N \quad \text{avec } P_0 = \infty \quad \text{et } A_N = \infty \quad (2.14)$$

En cherchant les invariants du kanban étendu on obtient des invariants semblables à ceux du kanban et du basestock :

$$A_{i-1} + F_i + P_i = K_i \quad \text{pour } i = 1 \dots N \quad (2.15)$$

$$D_{i-1} + F_i + P_i - D_i = S_i \quad \text{pour } i = 1 \dots N \quad (2.16)$$

Lorsque le nombre de kanbans est infini cette politique se comporte identiquement au basestock. Si dans chaque maille i le nombre de kanbans K_i est identique au niveau S_i qu'il faut maintenir alors le système est géré comme une politique kanban [DUR 1997]. La politique kanban étendu a un comportement assez proche de la politique kanban généralisé. Cependant, si pour chaque maille i , les paramètres K_i et S_i du kanban étendu sont identiques à ceux du kanban généralisé, alors le kanban étendu aura un temps de réponse à la demande plus petit que celui du kanban généralisé [DAL 1995].

2.5 Autres politiques de gestion de production

De nombreuses autres méthodes de gestion de production existent, et il serait difficile d'être exhaustif. Parmi toute cette diversité, on remarque cependant que certaines politiques sont parfois peu différentes les unes des autres. Certaines sont obtenues par combinaison de politiques simples, ce qui génère des politiques hybrides. Certaines politiques prennent en compte la production par lot ou des lignes de production avec assemblage.

2.5.1 Quelques variantes du Basestock

Les systèmes basestock comme toutes les politiques que nous avons vues jusqu'à présent, font partie des systèmes de type « production pour stock » car il produisent pour maintenir le niveau des stocks. Il existe aussi des systèmes de type « production pour ordre », ou « production à la commande », où l'on commence à fabriquer un produit seulement lorsque que la commande ferme qui lui est associée est arrivée. Dans ce dernier type de système il n'y a pas de stock de sécurité. Pour être de type « production à la commande », un système basestock doit avoir tous ses paramètres S_i nuls. Lorsqu'il n'y a que quelques niveaux S_i nuls

pour un système basestock donné, on parle alors parfois de système hybride entre une « production pour stock » et une « production à la commande » [LEE 1992].

Pour des systèmes à temps discret, le modèle de base du basestock est celui de [CLA 1960] dans lequel la capacité de production des processus de fabrication est infinie. Dans [FU 1994], [GLA 1994] et [GLA 1995] est introduit un système dans lequel la capacité de production des processus de fabrication est limitée, cette variante est alors appelé basestock modifié. La capacité de production, pour un système où le temps est discret, représente le nombre maximal de pièces pouvant être produites dans une unité de temps. Avec une politique basestock à chaque période les processus de fabrication doivent produire pour maintenir les stocks à leur niveau nominal. Une capacité de production qui n'est pas infinie va parfois empêcher cette politique de recouvrir un niveau de stock nominal en une seule période. Cette terminologie n'a donc pas d'utilité quand les temps sont continus. On remarque aussi que, lorsque le basestock est modifié, les résultats d'optimalité de [CLA 1960] ne s'appliquent plus.

Il existe aussi des politiques appelées (s,S) et (Q,r) qui sont deux politiques proches du basestock. Ces politiques sont classées en deux types appelés « type échelon » ou « type installation ». Ceci dépend des positions d'inventaire utilisées pour le contrôle de la production.

La position d'inventaire d'un échelon i est définie comme étant la somme des ordres de fabrication de la maille i et du nombre de produits présents dans les mailles i à N , moins le nombre de demandes de produits finis en attente.

La position d'inventaire d'une installation i est définie comme étant la somme des ordres de fabrication de la maille i et du nombre de produits présents dans les mailles i , moins le nombre de demandes de produits finis en attente.

Avec un système de production géré par la politique (s,S) un ordre de fabrication doit être lancé dans une maille i pour augmenter la position d'inventaire i jusqu'au niveau S_i chaque fois que celui-ci est passé en dessous du niveau s_i . Les paramètres s_i et S_i sont deux entiers qui vérifient $-\infty < s_i < S_i < +\infty$ [ZHE 1991].

Dans un système de production géré par la politique (Q,r) un ordre de fabrication pour Q_i pièces est transmis à la maille i chaque fois que la position de l'inventaire i est passé en dessous du niveau r_i .

Une politique (Q,r) de type installation peut être modélisée par un réseau de files d'attente (figure 2.9) en utilisant les notations des réseaux de Petri généralisés [DAV 1992]. Les barres obliques qui précèdent une station de synchronisation, indiquent que Q_i clients sont transformés en un unique client. A la sortie d'une station de synchronisation cette barre oblique indique par contre qu'un client est transformé en Q_i clients. Avec les noms utilisés pour les différents éléments du réseaux de files d'attente de la figure 2.9, la position d'inventaire d'une installation j est définie par :

$$i_j = Q_j.m'_j + F_j + P_j - Q_{j+1}.m'_{j+1}$$

pour j allant de 1 à N .

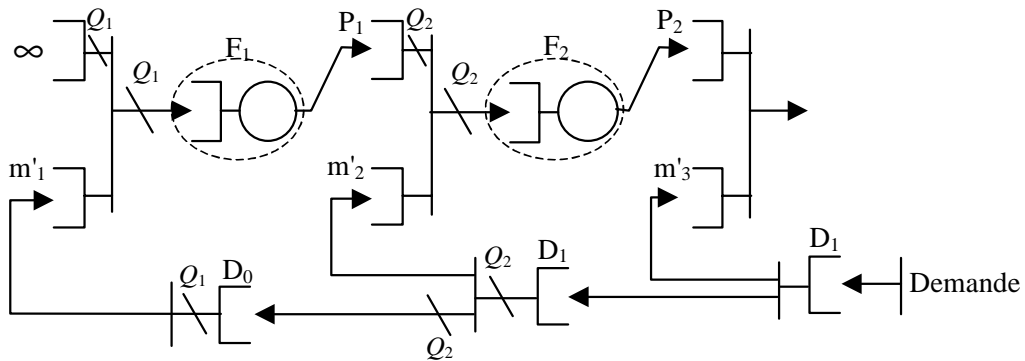


Figure 2.9 : Politique (Q,r) de type installation pour une ligne de deux mailles

De même on peut modéliser une politique (Q,r) de type échelon par réseaux de files d'attente (figure 2.10). Avec les noms utilisés pour les différents éléments du réseaux de files d'attente de la figure 2.10, la position d'inventaire d'un échelon j est définie par :

$$I_j = Q_j \cdot m'_j + \sum_{k=j}^N (F_k + P_k) - D_N$$

pour j allant de 1 à N .

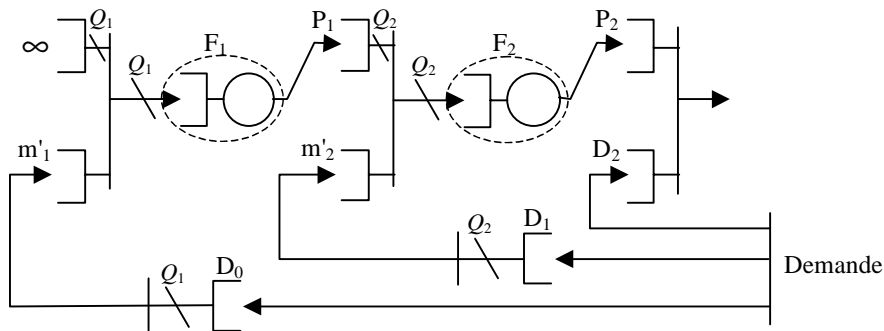


Figure 2.10 : Politique (Q,r) de type échelon pour une ligne de deux mailles

Si l'application des règles de contrôle des systèmes se fait de façon périodique ou si, à chaque instant, il est possible d'avoir plusieurs demandes de produits finis, alors les comportements de ces politiques (s,S) et (Q,r) sont différents. Par contre ces politiques peuvent être identiques quand le contrôle des processus de fabrication est effectué à chaque instant de façon continue dans le temps et si à chaque instant il y a au plus une demande d'un produit fini. Pour être identiques elles doivent alors vérifier $r_i = s_i$, $Q_i = S_i - s_i$ et avoir les mêmes positions d'inventaire initiales. Pour $s = S - 1$ on obtient la politique $(S-1,S)$ qui se comporte alors comme une politique basestock.

La politique «hedging point policy» consiste à maintenir les positions d'inventaire des échelons j à leur niveau Z_j appelé «hedging point». Elle peut se définir comme une politique (Q,r) de type échelon avec $Q_j=1$, et $r_j=Z_j-1$. Dans [LIB 2000] il est montré que cette

politique est équivalente au basestock dont l'objectif est de maintenir les positions d'inventaire des installations j à leur niveau S_j . Il suffit pour cela d'avoir $S_j = Z_j - Z_{j+1}$.

2.5.2 Quelques politiques avec production par lots

Les politiques (Q,r) sont, de par leur définition, faites pour gérer des systèmes avec production par lots. Dans l'article [AXS 1993] la différence entre les politiques (Q,r) de type installation et échelon est mise en évidence. Ils ont montré qu'on peut toujours trouver une politique (Q,r) de type échelon équivalente à une politique (Q,r) de type installation, l'inverse n'étant pas vrai. Leur conclusion est que la politique (Q,r) de type échelon est plus générale que celle de type installation. La différence de comportement entre ces deux types de politiques est minime. Avec une politique (Q,r) de type installation il y a toujours synchronisation de l'envoi d'un ordre de fabrication en amont avec ceux envoyés plus en aval. Cette synchronisation peut être évitée avec une politique (Q,r) de type échelon.

La politique basestock permet elle aussi de faire de la production par lots [BUZ 1993]. Le basestock réagit directement à la demande puisqu'il doit maintenir la position d'inventaire des échelons [BUZ 1992]. Le kanban peut lui aussi servir à la gestion de systèmes produisant par lots. Dans [BUZ 1993] les étiquettes kanban sont transmises en amont par paquets de taille identique aux lots.

2.5.3 Quelques variantes du kanban

Lorsque les processus de fabrication sont composés de plusieurs machines en série alors la politique kanban sera parfois appelée kanban segmenté [ETT 1995] puisque la ligne de production est alors découpée en segments. Pour [ETT 1995], le kanban correspond au cas où il y a une seule machine par maille. Dans [MUC 1995] ce type de politique est noté $N/(M_1, \dots, M_N) / (C_1, \dots, C_N)$ avec N le nombre de mailles, M_i le nombre de machines dans la maille i et C_i le nombre de kanban de la maille i . Dans cette thèse nous ne ferons pas cette distinction et appellerons kanban cette politique même si le système de production est composé de plusieurs machines.

Notons que lorsqu'il n'y a qu'une seule maille de production, cette politique sera aussi appelée kanban, même si dans de nombreux articles elle est souvent nommée CONWIP. Les initiales de cette politique sont celles de la formule « CONstant Work In Progress », ce qui signifie qu'elle a pour principe le maintien constant du nombre de pièces présentes dans la ligne de production. Cette politique a de plus l'avantage de réagir vite à la demande, puisque l'information remonte instantanément tant que le stock de produits finis n'est pas vide. Dans la thèse de A. Bonvik [BON 1996], cette politique a été comparée à d'autres politiques.

Quand les processus de fabrication ne sont composés que d'une machine non précédée d'une file d'attente tampon et ne servant qu'une pièce à la fois alors ceux-ci peuvent provoquer des

blocages. Quand une pièce ne peut pas passer d'une maille vers une autre, seulement à cause de ce type de blocage, alors la remontée de l'information est ralentie car un kanban n'aura pas été libéré à temps pour être transmis en amont. On appelle cette politique le kanban modifié [DIM 1990]. Elle est aussi parfois appelée "conventional kanban" par opposition au "minimum blocking kanban" [BON 1996] qui lui utilise des stocks tampons dans le processus de fabrication pour empêcher tout blocage.

Initialement, dans les années 70, le kanban pratiqué dans les usines de Toyota au Japon utilisait deux types d'étiquettes kanban. Une première catégorie d'étiquettes kanban, nommée P-kanban, circule en boucle dans le serveur et le stock de produits qui suit pour contrôler la production. Quant à l'autre type d'étiquettes kanban, nommé T-kanban, il contrôle le transport des pièces entre les mailles en circulant à l'intérieur des stocks tampon. On appelle cette politique le kanban bicarte ou "two-card kanban" [ETT 1995] par opposition au kanban monocarte ou "single-card kanban". Dans [DIM 1990], le kanban bicarte est considéré comme étant équivalent au kanban monocarte lorsque le transport de pièces est assimilé à une opération de production.

En [COT 1999], une modification de la politique kanban, appelée « kanban dynamique », a été développée. Cette politique utilise un mécanisme qui ralentit, lorsque cela est nécessaire, la remontée des étiquettes kanbans. Ceci a pour conséquence de limiter les encours, tout en conservant une qualité de service équivalente.

2.5.4 Politiques Finite-Buffer

La politique finite-buffer [ETT 1995] est une politique assez facile à mettre en œuvre. Elle consiste à produire tant que les stocks de produits intermédiaires ou finis ne sont pas pleins. La demande tire le flux de production en vidant les stocks. Lorsqu'un stock est plein le processus de fabrication qui précède se bloque. Il y a deux types de blocage, après service ou avant service. Si le blocage est avant service, les pièces ne rentreront pas dans le processus de fabrication si le stock qui suit est déjà plein. Quand le blocage est après service les pièces rentrent dans le processus de fabrication même si le stock qui suit est plein. Les pièces sont alors bloquées et elles empêchent que d'autres entrent dans le serveur tant que le stock suivant ne s'est pas vidé d'une unité. Les processus de fabrication n'utilisent pour leur contrôle que des informations locales et le nombre de pièces dans les mailles est limité. Pour ces deux raisons cette politique peut donc être considérée comme proche du kanban.

2.5.5 Quelques politiques hybrides

Les politiques kanban généralisé et kanban étendu sont deux politiques hybrides qui ont déjà été décrites dans les sections précédentes. Dans la thèse de Bonvik [BON 1996] plusieurs autres politiques hybrides sont proposées. Il propose principalement un hybride CONWIP/kanban modifié qu'il représente avec des cellules de contrôle, voir figure 2.11. Les

processus de fabrication ne possèdent pas de stock tampon, ils sont donc représentés par des cercles.

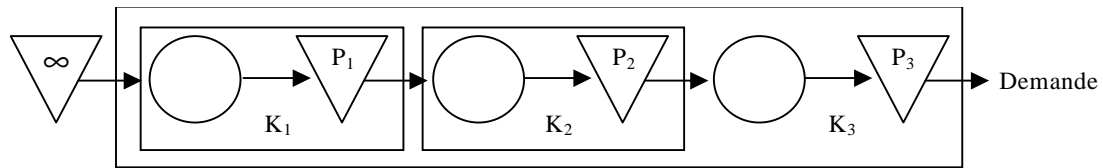


Figure 2.11 : Cellules de contrôle d'un CONWIP/kanban modifié pour trois mailles

Ce système comporte une boucle d'étiquettes kanban qui circulent entre la première et la dernière maille comme pour un CONWIP et des contrôles de type kanban à l'intérieur de toutes les mailles exceptée la dernière. Le nombre de pièces présentes dans cette dernière maille n'a pas besoin d'être contrôlé localement par une boucle d'étiquettes kanban car elle bénéficie déjà du contrôle d'encours effectué par le CONWIP.

Bonvik propose aussi un hybride CONWIP/kanban (stock tampon dans les processus de fabrication) et des hybrides basestock/kanban et basestock/kanban modifié. L'hybride basestock/kanban qu'il a étudié est similaire au kanban étendu mais pour simplifier l'analyse du système il a utilisé des niveaux de basestock nuls sauf pour le dernier étage. Un tel système ressemble alors à un hybride CONWIP/kanban car au lieu d'avoir une grande boucle kanban, il y a un unique renvoi d'information de type basestock entre le début et la fin de la ligne.

Dans [LIB 2000] de nombreuses politiques hybrides sont proposées. Parmi ces politiques des variantes de politiques connues sont obtenues seulement en ajoutant un contrôle local des encours dans les mailles (stage Work In Progress (WIP) control). Il s'agit pour une maille i de contrôler par une boucle d'étiquettes kanban les pièces présentes soit dans M_i (F_i privé de son stock tampon), soit dans F_i (avec stock tampon) ou bien dans M_i et P_i . Ces trois boucles d'étiquettes kanban sont respectivement nommées "manufacturing facility WIP (M-WIP) control", "input buffer and manufacturing facility WIP (IM-WIP) control" et "manufacturing facility and output buffer WIP (MO-WIP) control". Les figures 2.12 et 2.13 suivantes montrent deux exemples de ces politiques.

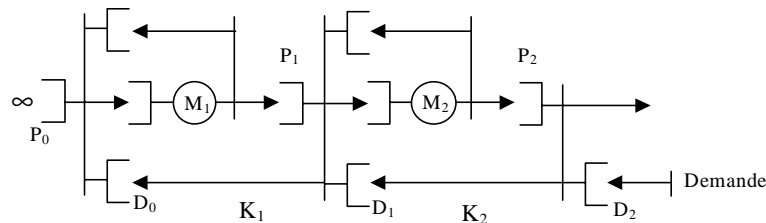


Figure 2.12 : Kanban avec contrôle IM-WIP pour une ligne de deux mailles

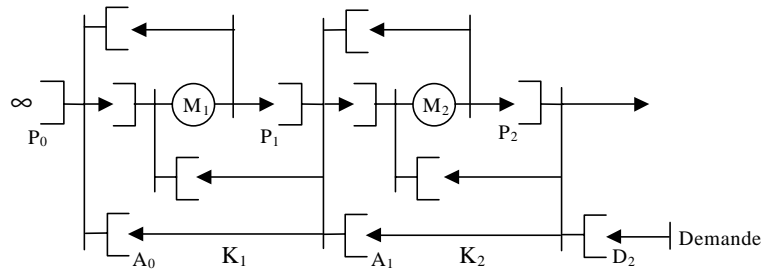


Figure 2.13 : Kanban avec contrôles IM-WIP et MO-WIP pour une ligne de deux mailles

Il est présenté, toujours dans [LIB 2000], des politiques avec des imbrications variées de boucles d'étiquettes kanban (figure 2.14).

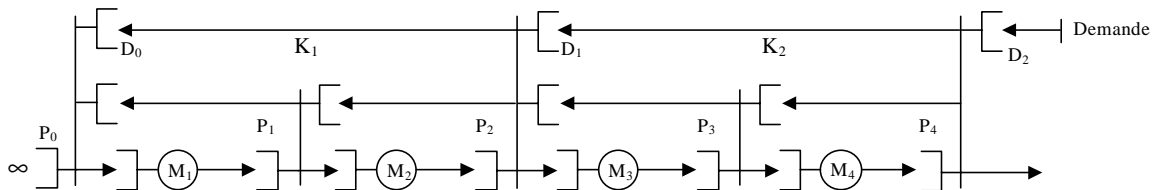


Figure 2.14 : Boucles kanban sur chaque maille et sur deux groupes de deux mailles en série.

D'autres politiques pourraient être présentées et en particulier des assemblages de politiques dues à la coopération d'entreprises qui n'utilisent pas les mêmes méthodes de gestion de production. On peut imaginer une entreprise qui utilise une politique kanban tandis que son fournisseur utilise le basestock, l'assemblage des deux donne alors une politique hybride.

Il existe aussi des politiques hybrides entre le flux poussé et le flux tiré. Le système PAC de Buzacott et Shanthikumar [BUZ 1993] est capable de modéliser plusieurs politiques à flux tiré mais il existe un paramètre qui introduit un délai dans la transmission des informations ce qui permet de réaliser un MRP.

2.5.6 Le MRP une politique à flux poussé

Le MRP (material requirement planning) est une politique qui a pour objectif de pousser au meilleur instant les produits bruts dans la ligne de fabrication. Il faut estimer le temps l mis pour produire un produit fini à partir de produits bruts. En fonction de ce temps et des prévisions sur les demandes à venir, il est possible de trouver l'instant de libération des pièces dans le système.

Les prévisions de la demande, les inventaires et les ordres de libération des pièces sont souvent fait à de mêmes instants périodiques [BUZ 1994] (discrete time MRP). Pour chaque période en fonction de la demande prévue et de l'état des stocks, on calcule la quantité de pièces qu'il faut envoyer dans le système de production. Par contre quand les prévisions sur la

demande et l'état du système sont connues de façon continue alors le contrôle du système peut être continu [BUZ 1989.b].

Il est souhaitable d'avoir une prévision de la demande faite sur une durée au moins égale à l , cependant cela n'est pas toujours le cas, il faut alors prévoir des stocks de sécurité. Les stocks de sécurité peuvent aussi être utilisés lorsque le temps de fabrication n'est pas parfaitement connu.

Suivant la qualité des prévisions sur la demande, il est parfois préférable de se contenter de maintenir à un niveau convenable les stocks de sécurité, plutôt que de faire des calculs utilisant le temps restant avant livraison [BUZ 1989.b]. Il y a donc un choix à faire entre le maintien de temps de sécurité et le maintien de stocks de sécurité [BUZ 1994].

Le basestock est d'ailleurs parfois considéré comme un cas particulier du MRP. Le système PAC de Buzacott et Shanthikumar permet de modéliser le basestock, et en ajoutant un mécanisme de temporisation de l'information il modélise aussi des systèmes à flux poussé comme le MRP. Les délais sont calculés en fonction des prévisions sur la demande et des temps de fabrication. L'idée d'introduire un délai est reprise dans [KAR 1999] où un système basestock avec temporisation des ordres de fabrication est faite. Cette temporisation est calculée en fonction de périodes de prévision fixes ou variables. Pour un MRP classique, elle ne peut être que fixe. (On voit ici que la frontière flux poussé flux tiré n'est pas toujours aussi nette).

Dans [BUZ 1989.a] un système kanban utilisant des prévisions de la demande, permet d'obtenir une gestion de production ayant beaucoup de similarités avec le MRP. Ainsi les avantages d'une production à flux poussé et à flux tiré sont cumulables.

2.5.7 Le pilotage optimal

Les politiques de pilotage des lignes de production consistent à contrôler chacun des processus de fabrication en fonction de l'état du système. En général l'objectif est de minimiser des coûts de possession et des coûts de non satisfaction des demandes. En général une commande de pilotage est optimale seulement si elle est de type "bang-bang". Cela signifie que les processus de fabrication peuvent être, suivant l'état du système, soit en marche avec un taux de service maximal, soit à l'arrêt avec un taux de service nul. Il a été montré que pour une ligne de fabrication de deux étages comportant des serveurs exponentiels la commande optimale pouvait être exprimée par des courbes de commandes possédant des propriétés particulières de monotonie [VEA 1992]. Dans [VEA 1994] des comparaisons entre des commandes optimales obtenues par programmation dynamique et des commandes équivalentes au basestock, au kanban et à d'autres politiques sont faites. Cette méthode de description des politiques sera à l'origine de la plupart des résultats exprimés dans cette thèse.

2.5.8 Quelques politiques pour des lignes avec assemblage

Considérons maintenant un système de production comportant une opération d'assemblage. En amont de cette opération, il y a deux lignes qui produisent deux produits intermédiaires différents. La figure montre un tel système de production composé de deux segments de lignes a et b suivis d'une station d'assemblage et éventuellement d'un dernier segment de ligne.

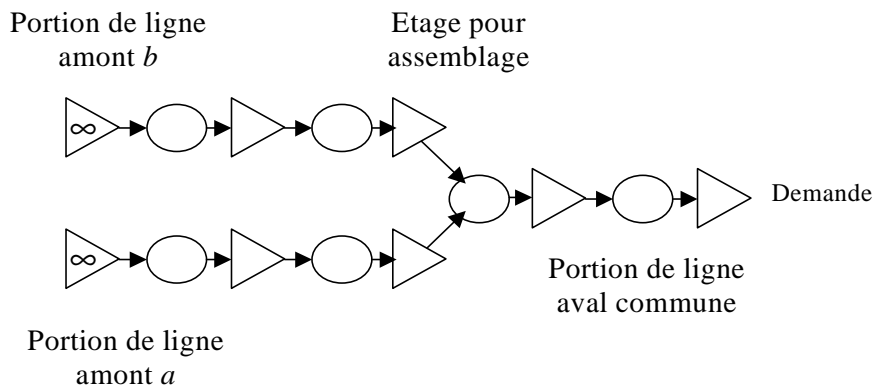


Figure 2.15 : Assemblage de deux produits

Le pilotage de tels systèmes peut se faire par une simple adaptation des politiques à flux tiré existantes. Dans [DIM 1996] une modélisation du kanban, figures 2.17 et 2.18, a été proposée pour des systèmes avec assemblage. Une modélisation de l'assemblage pour le kanban étendu a été proposée dans [CHA 2000]. Dans la thèse de Sbiti [SBI 2000] une modélisation de plusieurs politiques est proposée. La figure 2.16 montre une politique basestock avec une station d'assemblage.

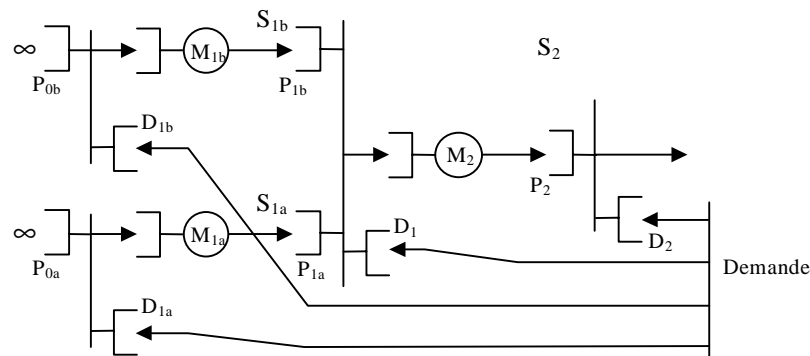


Figure 2.16 : Basestock avec une station d'assemblage

Pour le kanban et kanban généralisé plusieurs modélisations de l'assemblage sont possibles. L'assemblage peut se faire à libération simultanée ou à libération indépendante. Les deux types de libération pour une politique kanban avec assemblage sont illustrées sur la figure 2.17 dans le cas d'une libération simultanée et sur la figure 2.18 dans le cas d'une libération indépendante.

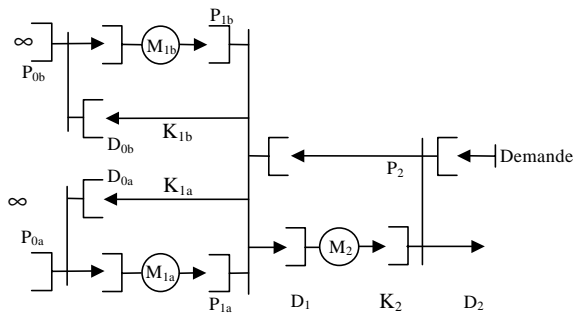


Figure 2.17 : Kanban avec assemblage à libération simultanée

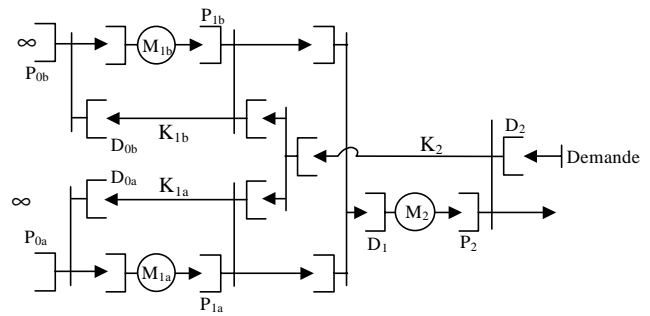


Figure 2.18 : Kanban avec assemblage à une libération indépendante

Dans [SBI 1999] il est proposé pour le kanban généralisé une modélisation d'un assemblage à libération simultanée, noté KGS (voir figure 2.19). Un kanban généralisé avec assemblage à libération indépendante, noté KGI, y est aussi proposé ; nous ne l'avons pas illustré par un réseau de files d'attente car nous avons préféré montrer une version plus simple (figure 2.20) qui reste cependant équivalente au KGI (cela est montré dans [SBI 1999]).

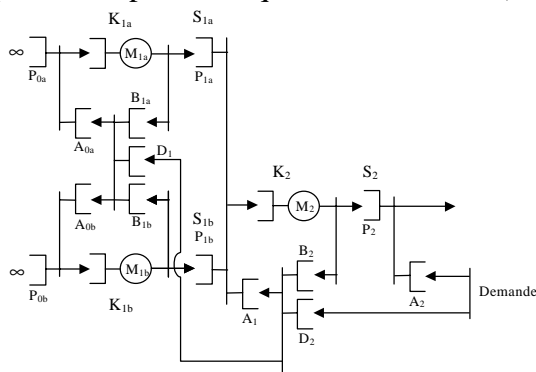


Figure 2.19 : Kanban généralisé avec assemblage à libération simultanée (KGS)

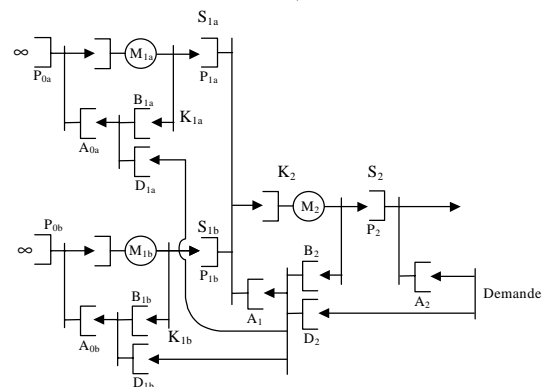


Figure 2.20 : Kanban généralisé avec assemblage équivalent à une libération indépendante (KGI équivalent)

Il a été montré que les systèmes KGI se comportaient identiquement ou mieux que les systèmes KGS [SBI 1999].

2.6 Conclusion

Les politiques de gestion de production sont nombreuses. Certaines d'entre elles présentent des similitudes. Ces politiques ont des formulations souvent différentes qui ne simplifient pas leur compréhension, leur analyse et leur comparaison.

Nous avons donc proposé dans le chapitre suivant une formulation unifiée des politiques à flux tiré. Cette représentation est utilisée par la suite dans chacun des autres chapitres.

Eléments pour une formulation unifiée de la dynamique des politiques à flux tiré

Dans ce chapitre nous ne traiterons que des systèmes de production gérés à flux tiré. Comme nous l'avons montré dans le chapitre 2, différentes méthodes de gestion des flux de production existent, on les appelle aussi politiques de gestion de production et par abus de langage simplement politiques. Lorsqu'une politique de gestion de production est appliquée à un système de production il est alors possible de connaître précisément de quelle façon va évoluer chaque état du système. La définition des politiques doit donc être très précise et, pour définir celles-ci, différentes méthodes de formulation sont possibles. L'objectif de ce chapitre sera de montrer une nouvelle technique de formulation des politiques et de mettre en évidence l'intérêt de celle-ci.

Pour formuler une politique, on peut utiliser les réseaux de files d'attente, ou tout simplement une définition des mécanismes correspondants à la mise en œuvre réelle du système de production. La formulation qui est présentée dans ce chapitre est plus abstraite car elle utilise des fonctions permettant de caractériser la politique.

Pour un certain nombre de politiques ces fonctions ont la même structure, seules quelques paramètres peuvent changer. Au contraire de représentations graphiques, comme celle faites par des réseaux de files d'attente, cette formulation n'a pas pour principal objectif une compréhension intuitive des politiques, cependant elle permet de mettre en œuvre des techniques qui peuvent être plus efficaces que celles utilisées auparavant. Nous montrerons par exemple dans le prochain chapitre que de telles formulations permettent de trouver facilement toute équivalence de comportement existant entre deux politiques.

Cette formulation nous a été inspirée par un article de Veatch et Wein [VEA 1994] qui traite des problèmes de politique optimale sur une ligne de deux étages. Ces politiques optimales, décrites par des courbes de commande, sont comparées à d'autres politiques dites sous-optimales telles que le basestock ou le kanban. Pour utiliser des méthodes de calcul similaires, des politiques sous-optimales ont elles aussi été traduites sous forme de courbes de commande. Cependant, dans cet article, Veatch et Wein n'ont traité que des lignes de deux

stations monoserveurs gérées par une politique basestock, kanban ou fixed-buffer. C'est pour cette raison que nous avons généralisé les calculs pour obtenir une formulation plus complète. Ainsi avec cette nouvelle formulation, il est possible de décrire des systèmes ayant un nombre quelconque d'étages avec des processus de fabrication pratiquement quelconques. La gestion du système peut se faire en kanban, basestock ou par d'autres politiques dont nous parlerons plus loin. Il est aussi possible d'utiliser cette formulation, avec un minimum de modifications, pour des lignes de production gérées par lots ou pour des systèmes comportant des ateliers d'assemblage.

Pour expliquer ces résultats, il est nécessaire de décrire les éléments de base utiles à la formulation des politiques (section 3.1). Nous allons définir des variables d'état du système et nous montrerons dans les sections 3.2 et 3.3 que la dynamique du système est connue lorsque le nombre de pièces présentes dans les processus de fabrication est exprimé en fonction de ces variables.

Dans le chapitre 4, le nombre F_i de pièces contenues dans les processus de fabrication F_i sera calculé pour chaque politique en fonction de variables d'état. Ces calculs seront généralisés dans la section 4.3 à une catégorie très générale de politiques à flux tiré. On montrera comment utiliser ces calculs en fonction du type de processus de fabrication employé. Les valeurs F_i seront aussi calculées pour des systèmes produisant par lots et pour des systèmes avec station d'assemblage. Cela permettra dans le chapitre 5 d'effectuer des comparaisons de politiques à l'aide de cette formulation.

3.1 Notations et éléments nécessaires à l'expression de notre formulation.

Cette formulation a pour objectif de décrire le comportement du système. Pour cela il est nécessaire de définir les éléments constitutifs du système. Nous montrerons ensuite quelles sont les variables d'état qui ont été utilisées pour concevoir cette formulation.

Dans les sections suivantes, ces éléments nous permettront de proposer une définition de la dynamique d'un système de production. A partir de cette définition, nous montrerons pourquoi un simple calcul du nombre de pièces présentes dans le processus de fabrication est suffisant pour caractériser la dynamique du système de production.

Notons que l'ensemble des notations est résumé dans l'annexe A.

3.1.1 Description détaillée des mailles et des processus de fabrication

Le système considéré est une ligne de production composée de N mailles (figure 3.1).

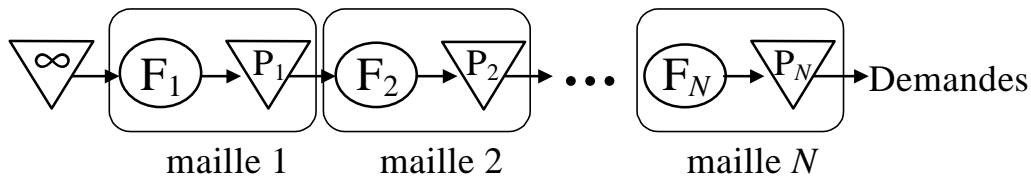


Figure 3.1 : Ligne de fabrication composée de N mailles

En amont il y a un stock de produits bruts et en aval des demandes de produits finis. Les demandes non satisfaites sont mises en attente et leur nombre est D_N . Une maille i est composée d'un processus de fabrication F_i et d'un stock P_i où les pièces attendent avant de passer à l'atelier suivant. Les variables F_i et P_i sont les nombres respectifs de pièces contenues dans F_i et P_i . Ce découpage en étages est classique car, habituellement, les politiques de gestion de production sont définies par des règles définissant le passage des pièces d'une maille à une autre.

Dans le cas général, un processus de fabrication peut être relativement complexe et comporter plusieurs stations. Il peut être précédé d'une file d'attente tampon mais cela n'est pas toujours le cas. Lorsqu'il n'y a pas de file d'attente tampon, il peut se produire un blocage des pièces arrivant de P_i . Ceci est alors dû à la non disponibilité des ressources du processus de fabrication F_i . Dans ce cas, à un instant donné, on observe que les processus de fabrication ne peuvent pas accepter à leur entrée plus qu'un certain nombre de pièces. Ce nombre sera noté C_i pour un processus de fabrication F_i . Il peut être fini ou non et sa valeur peut varier dans le temps suivant l'état interne de F_i .

Les figures 3.2 et 3.3 montrent deux exemples de processus de fabrication précédés d'un stock avec leurs capacités d'accueil C_i respectives.

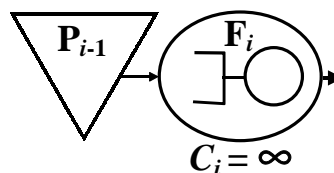


Figure 3.2 : Capacité d'accueil d'un processus de fabrication composé d'une station monoserveur précédée d'une file d'attente tampon

La présence d'une file d'attente tampon en début de processus de fabrication autorise à chaque instant l'entrée d'un nombre quelconque de pièces. On a donc $C_i = \infty$ pour l'exemple de la figure 3.2. Pour l'exemple de la figure 3.3, deux valeurs peuvent être prise par C_i . Lorsque la première station du processus de fabrication est occupée on a $C_i = 0$ sinon $C_i = 1$.



Figure 3.3 : Capacité d'accueil d'un processus de fabrication, composé d'une série de deux stations monoserveur, non précédé d'une file d'attente tampon

La figure 3.4 montre en détail la modélisation par réseau de Petri d'un stock suivi d'un processus de fabrication quelconque. L'entrée des pièces dans le processus de fabrication dépend du contrôle effectué par la politique mais il dépend aussi de la valeur de C_i . Lorsque cette valeur est nulle, on ne peut pas faire rentrer de pièce dans F_i . La sortie des pièces de F_i ne dépend que de la venue des événements u_{F_i} ou u_{FC_i} . L'événement u_{FC_i} fait sortir une pièce et rétablit une place d'accueil supplémentaire puisque l'entrée de la pièce en avait consommé une. Ce type d'événement se produit, par exemple, quand le processus de fabrication n'est constitué que d'un ou plusieurs serveurs en parallèle. Par contre pour un système où les processus de fabrication sont constitués de plusieurs monoserveurs en série, il se produira des événements de type u_{F_i} et u_{C_i} . L'événement u_{F_i} fait sortir une pièce de F_i mais ne rétablit pas de place d'accueil, il faut attendre l'événement u_{C_i} pour qu'une d'elles soit rétablie.

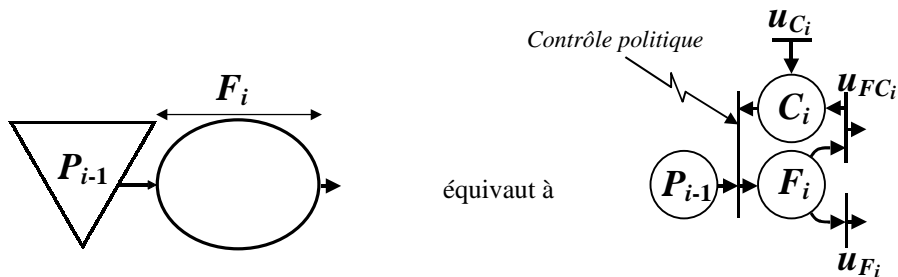


Figure 3.4 : Eléments d'une maille avec un processus de fabrication quelconque

Les processus de fabrication F_i considérés pour un étage i seront souvent composés d'un stock tampon de capacité infinie suivi d'un mécanisme de production quelconque nommé M_i . Ainsi, dans ce cas, la valeur C_i sera toujours infinie. Ceci permet d'accueillir des pièces dans F_i même si M_i est indisponible, on évite alors tout blocage en amont de F_i .

Il sera parfois utile de s'intéresser à C'_i la capacité d'accueil de M_i . Par ailleurs, certaines politiques ont tendance à vider les stocks P_{i-1} dans les stocks tampon des processus de fabrication F_i . Il est donc intéressant d'utiliser l'ensemble stock P_{i-1} plus le stock tampon du processus de fabrication F_i qui sera noté P'_{i-1} plutôt que d'utiliser P_{i-1} . Le nombre de pièces contenues en P'_i est noté P'_i . Par extension le stock de produits finis sera aussi noté P'_N , mais on a $P'_N = P_N$.

La figure 3.5 montre en détail la modélisation par réseau de Petri d'un processus de fabrication avec stock tampon infini. Comme pour le cas d'une maille avec processus de fabrication quelconque, représenté figure 3.4, il y a trois types d'événements. Les événements u_{M_i} et $u_{MC'_i}$ font sortir une pièce du mécanisme de fabrication et le rétablissement d'une place d'accueil s'effectue suite à $u_{MC'_i}$ et $u_{C'_i}$ par incrémentation de C'_i .

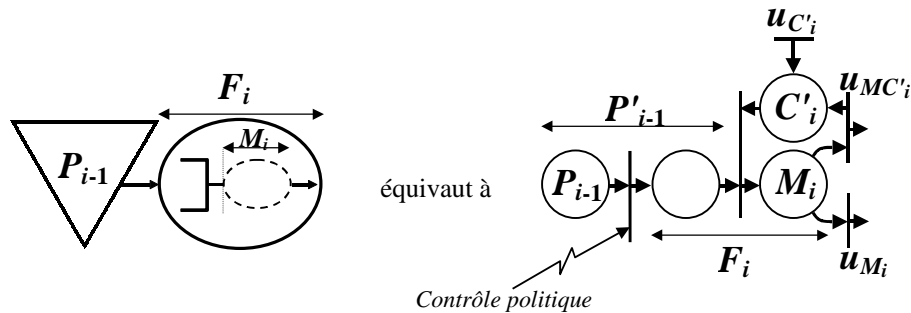


Figure 3.5 : Processus de fabrication quelconque avec stock tampon infini. Les pièces sortent une à une et C'_i est augmenté au plus de un par événement.

Pour les systèmes considérés, on supposera que la demande n'est jamais perdue et qu'elle est servie immédiatement tant qu'il reste des produits finis. Ainsi lorsque l'événement u_D , qui correspond à une demande, se produit, alors P_N diminue de un s'il est strictement positif, sinon D_N augmente de un. Au moins une des quantités P_N ou D_N est nulle, on a donc :

$$\min(P_N, D_N) = \min(P'_N, D_N) = 0 \quad (3.1)$$

Après une description détaillée des processus de fabrication, il est possible de décrire l'état du système avec seulement une variable par maille. Nous avons appelé ces variables des variables d'état global.

3.1.2 Description de variables dites "d'état global"

Les variables présentées ici sont appelées variables d'état global car pour certains cas elles suffisent à décrire l'état du système même si plusieurs valeurs décrivant le système ont été agrégées. Leur définition est une simple extension de celle utilisée en [VEA 1994].

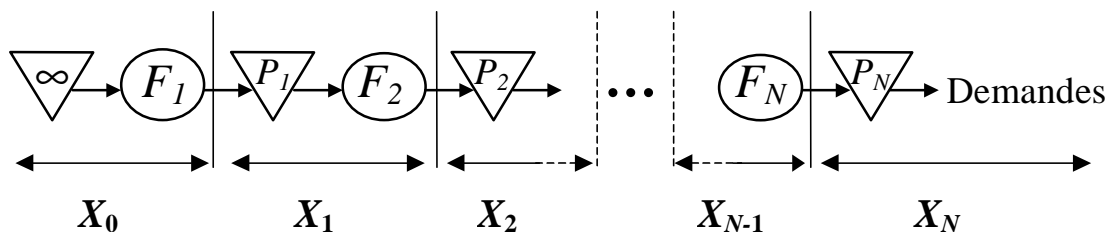


Figure 3.6 : Variables d'état global pour une ligne de N mailles

Pour définir nos variables d'état global, on effectue un découpage de la ligne qui passe entre le processus de fabrication et le stock de chaque étage (figure 3.6). Ainsi, les variables X_i

pour i allant de 1 à $N-1$ ont pour valeur $(P_i + F_{i+1})$ qui est la somme des pièces présentes dans le stock de l'étage i et des pièces présentes dans le processus de fabrication de l'étage $i+1$. La variable X_N est la somme des pièces présentes dans le stock de l'étage N moins le nombre de demandes en attente. On a donc les formules suivantes pour le vecteur d'état global X :

$$X_0 = P_0 + F_1 = \infty$$

$$X_i = P_i + F_{i+1} = P'_i + M_{i+1} \quad \text{pour } i \text{ allant de } 1 \text{ à } N - 1 \quad (3.2)$$

$$\text{et } X_N = P_N - D_N = P'_N - D_N \quad (3.3)$$

La variable X_0 ne contient pas d'information sur l'état du système car sa valeur est toujours infinie. On notera 1_i le vecteur qui est nul partout sauf pour sa $i^{\text{ème}}$ composante qui vaut un.

Dans les sections suivantes nous montrons que sous certaines hypothèses, il est possible de déterminer l'état du système en fonction du seul vecteur X . Sous certaines hypothèses nous montrons qu'il est aussi possible de déterminer le comportement du système par la seule détermination des valeurs F_i en fonction de X .

Dans la section 3.2, on s'intéresse à des lignes de production où le processus de fabrication est constitué d'une file d'attente infinie (le stock tampon infini) suivie d'un monoserveur. Dans la section 3.3, on s'intéresse à d'autres types de processus de fabrication.

3.2 Dynamique et formulation des politiques pour une ligne avec stations monoserveur précédées d'un stock tampon.

Pour simplifier cet exposé, nous commencerons par traiter le cas particulier où les processus de fabrication sont composés d'une station monoserveur précédée d'un stock tampon de capacité infinie. La figure 3.7 montre le comportement d'un tel processus de fabrication. Les seuls événements possibles sont u_{F_i} et $u_{MC'_i}$.

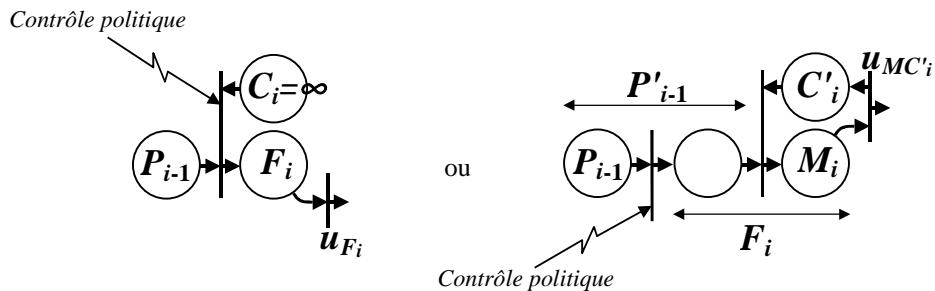


Figure 3.7 : Processus de fabrication avec station monoserveur et stock tampon

Ce type de processus de fabrication vérifie alors les propriétés suivantes.

$$C_i = \infty, C'_i + M_i = 1 \quad \text{et} \quad M_i = \min(1, F_i) \quad (3.4)$$

La capacité d'accueil C_i est infinie à cause du stock tampon. Par contre, C'_i la capacité d'accueil de M_i vaut un si la station monoserveur est vide et zéro sinon. Le nombre M_i de pièces présentes dans le processus de fabrication privé de son stock tampon vaut zéro si F_i est nul et un sinon.

3.2.1 Définition de la notion de dynamique

Maintenant que le type de processus de fabrication utilisé a été explicité, nous allons donner une définition de la notion de dynamique dans une ligne de fabrication. Ceci nous permettra ensuite de montrer que la formulation des politiques qui sera proposée, détermine sans ambiguïté la dynamique des systèmes de production.

3.2.1.1 Influence des critères de comparaison sur une définition de la dynamique

Nous utiliserons deux définitions de la dynamique, car suivant les critères de comparaison employés, les grandeurs à prendre en compte pour définir l'état du système ne sont pas les mêmes. Deux types de critères seront définis et pour chacun nous associerons une certaine définition de la dynamique.

Le premier critère, dit de type (a), fait la différence entre une pièce placée dans le stock P_{i-1} de la maille $i-1$ et une autre pièce qui aurait été poussée dans la maille i suivante, même si celle-ci est restée dans le stock tampon de F_i sans être encore passée dans M_i . Par contre, ce critère ne fait pas de différence entre une pièce en attente d'un service dans le stock tampon de F_i et une autre qui serait en service dans M_i .

Le second critère, dit de type (b), a le comportement inverse. Il prend en compte de la même façon les contenus juxtaposés du stock P_{i-1} et du stock tampon de F_i qui sont compris entre M_{i-1} et M_i . Ce critère considère différemment une pièce placée dans le stock tampon de F_i et une autre placée en M_i . Ce qui importe ici, est l'exécution effective de services.

Ces deux critères prennent chacun en compte le nombre des demandes non servies. Ainsi, pour un critère de type (a), la dynamique sera définie en utilisant les grandeurs P_i , F_i et D_N tandis qu'avec un critère de type (b) il faudra utiliser les quantités P'_i , M_i et D_N .

3.2.2 Définition de la notion de dynamique pour un critère de type (a)

Des systèmes définis par des méthodes différentes peuvent avoir un comportement identique. Il faut donc identifier clairement le comportement des systèmes de production. Pour atteindre cet objectif nous allons définir la notion de dynamique d'un système de production. Il s'agit en fait de définir tous les états atteignables du système, les événements possibles pour chacun de ces états et le changement d'état en fonction de l'événement. Cependant pour que le système décrit soit bien un système de production, il devra respecter un certain nombre de propriétés. Nous nous limiterons dans cette section à la notion de dynamique pour un critère de type (a) dans des lignes de production dont les processus de

fabrication sont composés d'un stock tampon et d'un monoserveur. Avant de donner une définition de cette dynamique nous allons présenter certaines notations utilisées (voir la liste récapitulative en annexe A).

Il faut commencer par définir quelles sont les valeurs caractéristiques du système qui vont définir son état. Le choix de ces valeurs est fait en fonction de leur influence sur la dynamique du système et des critères utilisés. On notera E l'ensemble des vecteurs d'états admissibles pour ce type de système de production considéré. Chaque vecteur e de E a pour composantes les $3N+1$ valeurs suivantes $D_N, P_1, \dots, P_N, F_1, \dots, F_N, C_1, \dots, C_N$. Pour le cas considéré dans toute la section 3.2, les composantes de valeur C_i sont toujours infinies et elles ne sont donc pas vraiment utiles. Cependant pour rester cohérent avec des cas plus généraux ces dernières composantes sont conservées. On notera e_{D_N} , e_{P_i} , e_{F_i} et e_{C_i} les composantes respectives D_N , P_i , F_i et C_i de chaque état $e \in E$. Ainsi la valeur e_{D_N} est la 1^{ière} composante du vecteur e , e_{P_i} est la $(1+i)$ ^{ième} composante de e , e_{F_i} est la $(1+N+i)$ ^{ième} composante de e et e_{C_i} est la $(1+2N+i)$ ^{ième} composante de e pour i allant de 1 à N .

Pour chaque état admissible, il est possible d'associer un certain nombre d'événements. Suite à l'arrivée de ces événements l'état du système change. Pour définir la dynamique des lignes de fabrication, nous utiliserons donc une application D qui à chaque couple (état initial, événement) va associer un état final.

On note EU l'ensemble des couples composés d'un état de E et d'un événement. Pour chaque état e de E , on note U_e l'ensemble des événements pouvant se produire lorsque l'état du système est e . Les types d'événements pouvant se produire dans le cas étudié sont u_D et u_{F_i} . Le premier correspond à l'arrivée d'une demande et le suivant correspond à une pièce qui sort du processus de fabrication i .

A chaque instant, il se produit au plus un événement. Les demandes arrivent indépendamment de l'état du système donc u_D sera toujours élément de U_e . Tant qu'il y a des pièces présentes dans un processus de fabrication F_i , u_{F_i} peut se produire. Ainsi pour un état e donné, u_{F_i} pourra se produire si et seulement si $e_{F_i} > 0$. L'événement nul est défini par \emptyset , il sera le seul à ne pas changer l'état du système. Tout cela permet d'écrire la définition qui suit.

Définition 3.1.a :

Pour une ligne de production dont les processus de fabrication sont composés d'un stock tampon suivi d'une station monoserveur, on définit les événements discrets pouvant se produire dans un état e de E , par l'ensemble :

$$U_e = \{\emptyset, u_D\} \cup \{u_{F_i} \mid e_{F_i} > 0, i \in \{1, \dots, N\}\}$$

□

A partir d'un état e de E , il est toujours possible de calculer l'état global X (défini précédemment en section 3.1.2) qui lui est associé. Ainsi nous définissons ci-dessous une application \mathfrak{X} qui permet de passer d'un vecteur d'état e de E , comportant $3N+1$ composantes, vers un vecteur d'état global X qui ne comporte que N composantes. Une des questions importante de ce chapitre sera de vérifier que, malgré une différence de leur dimension respective, les vecteurs d'état global X et les vecteurs d'état e de E sont équivalents. Pour prouver cela il sera nécessaire de trouver application bijective définie de E vers l'ensemble des vecteurs d'état global X admissibles.

Définition 3.2.a :

On définit l'application \mathfrak{X} par :

$$\mathfrak{X} : \quad E \longrightarrow \mathfrak{X}[E]$$

$$e = (e_{D_N}, e_{P_1}, \dots, e_{P_N}, e_{F_1}, \dots, e_{F_N}, e_{C_1}, \dots, e_{C_N}) \mapsto X = (e_{P_1} + e_{F_2}, \dots, e_{P_{N-1}} + e_{F_N}, e_{P_N} - e_{D_N})$$

□

Nous allons maintenant donner une définition formelle de la dynamique d'une ligne de production en utilisant toutes les notations précédemment définies.

Définition 3.3.a :

La dynamique de type (a) (c'est-à-dire vue au niveau des stocks et processus de fabrication), pour une ligne de production de n étages avec des processus de fabrication composés d'un stock tampon infini suivi d'un monoserveur, est définie par une application D vérifiant :

$$D : EU \longrightarrow E \quad \text{avec} \quad E \subset \mathbb{N}^{1+2N} \times \{\infty\}^N$$

$$(e, u) \mapsto f \quad EU = \{(e, u) \mid e \in E; u \in U_e\}$$

où e est l'état du système à l'instant qui précède l'arrivée d'un événement u et f est l'état du système après son arrivée. L'ensemble U_e est décrit dans la définition 3.1.a . Pour tout état e élément de E , les propriétés suivantes doivent être vérifiées :

$$\min(e_D, e_{P_N}) = 0 \quad (3.5)$$

$$D(e, \emptyset) = e \quad (3.6)$$

$$\mathfrak{X}(D(e, u_D)) = \mathfrak{X}(e) - 1_N \quad (3.7)$$

$$u_{F_1} \in U_e \Rightarrow \mathfrak{X}(D(e, u_{F_1})) = \mathfrak{X}(e) + 1_1 \quad (3.8)$$

$$\forall i \in \{2, \dots, N\}, \quad u_{F_i} \in U_e \Rightarrow \mathfrak{X}(D(e, u_{F_i})) = \mathfrak{X}(e) + 1_i - 1_{i-1}$$

□

Chaque vecteur d'état élément de E possède $1+2N$ composantes à valeurs entières plus N composantes infinies car ici la capacité d'accueil est toujours infinie.

La propriété (3.5) montre qu'il ne peut il y avoir simultanément des demandes non servies et des produits finis. En utilisant l'équation (3.6) on montre que l'état ne doit pas changer en présence d'un événement nul \emptyset .

La formule (3.7) montre que l'inventaire des produits finis décroît d'une unité (-1_N) en présence d'une demande (événement u_D) tandis que les inventaires X_i restent constants.

Les formules en (3.8) montrent que u_{F_i} , l'événement correspondant à une pièce qui sort de F_i , fait croître d'une unité (1_i) l'inventaire X_i compris entre les sorties de F_i et F_{i+1} . De plus, cet événement u_{F_i} fait décroître X_{i-1} d'une unité (1_{i-1}) lorsque $i > 1$.

La détermination du domaine de définition EU de l'application D permet de connaître les états atteignables du système. Si deux systèmes A et B sont dans le même état à l'instant initial, s'ils ont des distributions du temps de service et des arrivées de demandes identiques et s'ils ont une même dynamique alors pour tout instant t chaque état a une probabilité en A identique à celle en B.

Il est donc possible de comparer le comportement de deux systèmes dont les processus de fabrication ont un stock tampon et une station monoserveur. Pour faire de telles comparaisons on supposera implicitement que les deux systèmes ont un même état initial et que les distributions sont identiques. Ainsi il ne reste alors plus qu'à comparer l'identité des dynamiques au sens de la définition 3.3.a. Pour que deux dynamiques soient identiques, il suffit que pour chacune des lignes, les applications D et les ensembles EU de la définition 3.3.a soient identiques.

3.2.3 Définition de la notion de dynamique pour un critère de type (b)

Pour un critère de type (b) les valeurs pertinentes sont D_N , M_i , P'_i , et C'_i . Il faut donc modifier la définition 3.3.a en conséquence :

On note E' l'ensemble des vecteurs d'états admissibles pour les critères de type (b) avec le système de production considéré. Chaque vecteur de E' a pour seules composantes la séquence des valeurs suivantes $D_N, P'_1, \dots, P'_N, M_1, \dots, M_N, C'_1, \dots, C'_N$. Ces composantes sont respectivement notées e_{D_N} , $e_{P'_i}$, e_{M_i} et $e_{C'_i}$.

On note $E'U'$ l'ensemble des couples composés d'un état de E' et d'un événement. Pour chaque état e de E' on note U'_e l'ensemble des événements discrets pouvant se produire. Les événements pouvant se produire, dans le cas étudié avec ce type de critère, sont u_D et $u_{MC'_i}$. L'événement $u_{MC'_i}$ ne peut se produire que si M_i est supérieur à zéro ; il correspond à une pièce qui sort de M_i et qui simultanément crée une augmentation de la capacité d'accueil.

Définition 3.1.b :

Pour une ligne de production dont les processus de fabrication sont composés d'un stock tampon suivi d'une station monoserveur, on définit les événements discrets pouvant se produire dans un état e de E' , par l'ensemble :

$$U'_e = \{\emptyset, u_D\} \cup \{u_{MC'_i} \mid e_{M_i} > 0, i \in \{1, \dots, N\}\}$$

□

On peut définir l'application suivante pour obtenir en fonction d'un état donné, l'état global correspondant.

Définition 3.2.b :

On définit l'application \mathfrak{X}' par :

$$\mathfrak{X}' : \quad E' \longrightarrow \mathfrak{X}'[E']$$

$$e = \left(e_{D_N}, e_{P'_1}, \dots, e_{P'_N}, e_{M_1}, \dots, e_{M_N}, e_{C'_1}, \dots, e_{C'_N} \right) \mapsto X = \left(e_{P'_1} + e_{M_2}, \dots, e_{P'_{N-1}} + e_{M_N}, e_{P'_N} - e_{D_N} \right)$$

□

On peut ainsi écrire la définition suivante d'une dynamique pour un critère de type (b).

Définition 3.3.b :

La dynamique de type (b) (c'est-à-dire vue au niveau de ses stocks plus stocks tampon et de ses processus de fabrication sans stock tampon), pour une ligne de production de n étages avec des processus de fabrication composés d'un stock tampon infini suivi d'un monoserveur, est définie par une application D' vérifiant :

$$D' : E'U' \longrightarrow E' \quad \text{avec} \quad E' \subset \mathbb{N}^{1+2N} \times \{0;1\}^N$$

$$(e, u) \mapsto f \quad E'U' = \{(e, u) \mid e \in E'; u \in U'_e\}$$

où e est l'état du système à l'instant qui précède l'arrivée d'un événement u et f est l'état du système après son arrivée. L'ensemble U'_e est décrit dans la définition 3.1.b. Pour tout e élément de E' , e et D' doivent vérifier les propriétés suivantes :

$$\min(e_D, e_{P'_N}) = 0 \quad (3.9)$$

$$\forall i \in \{1, \dots, N\}, \quad e_{M_i} + e_{C'_i} = 1 \quad (3.10)$$

$$D'(e, \emptyset) = e \quad (3.11)$$

$$\mathfrak{X}'(D'(e, u_D)) = \mathfrak{X}'(e) - 1_N \quad (3.12)$$

$$u_{MC'_i} \in U'_e \Rightarrow \mathfrak{X}'(D'(e, u_{MC'_i})) = \mathfrak{X}'(e) + 1_1 \quad (3.13)$$

$$\forall i \in \{2; \dots; N\}, \quad u_{MC'_i} \in U'_e \Rightarrow \mathfrak{X}'(D'(e, u_{MC'_i})) = \mathfrak{X}'(e) + 1_i - 1_{i-1}$$

□

Les définitions 3.3.a et 3.3.b sont très proches. On peut cependant remarquer l'équation (3.10) qui n'a pas d'équivalent dans la définition 3.3.a. Cette équation correspond à une des propriétés (3.4) déjà vue précédemment. Le serveur M_i peut donc soit contenir une pièce en service et avoir une capacité d'accueil nulle, ou bien avoir une capacité d'accueil de un et ne contenir aucune pièce.

Dans les définitions 3.3.a et 3.3.b, les ensembles E et E' ont un rôle important dans la détermination de la politique de gestion de production, car l'arrivée des événements u_{F_i} et

u_{MC_i} dépendent respectivement de e_{F_i} et e_{M_i} . Par exemple si e_{F_i} est nul alors F_i est vide, il n'y a pas de pièce en service et u_{F_i} ne peut pas se produire.

Les applications D et D' jouent aussi un rôle important en déterminant les changements d'état. Cependant, le nombre d'états possibles à la suite de l'occurrence d'un événement est limité par les propriétés des définitions 3.3.a et 3.3.b et aussi par les vecteurs d'état en E et E' . Les différents états possibles permettent de choisir le nombre de pièces à libérer dans F_i ou M_i . Les ensembles E et E' sont tels qu'à la suite d'un événement il n'y a qu'un seul nombre de pièces ayant pu être libéré dans F_i ou M_i . Ainsi on en déduit les fonctions D et D' car pour chaque élément de EU et $E'U'$ ces fonctions ne peuvent avoir qu'une seule valeur. Dans certains cas il est donc possible, avec un minimum d'information, de déterminer la dynamique du système.

3.2.4 Détermination de la dynamique par F_i ou M_i en fonction de X

Pour toute politique, la gestion de la production s'effectue par un contrôle direct ou indirect de la marche et de l'arrêt des stations dans les processus de fabrication. Dans notre cas les processus de fabrication fonctionnent à flux poussé. Ils sont en service dès que des pièces y sont introduites. Les stocks intermédiaires ne sont là que pour faire attendre les pièces qui ne peuvent pas, ou ne doivent pas, passer sur une machine.

Une fois qu'une pièce est entrée dans un processus de fabrication, il n'y a plus aucun contrôle possible sur celle-ci, tant qu'elle n'est pas sortie pour aller dans le stock qui suit. Le contrôle de la production ne peut s'effectuer que par des blocages du flux de production à l'entrée des processus de fabrication. La dynamique des systèmes de production va dépendre de ces contrôles qui indirectement vont agir sur le fonctionnement des machines.

Nous montrons en section 3.2.4.1, pour un critère de type (a), que la connaissance du flux de pièces autorisées à entrer dans les processus de fabrication F_i peut se déduire de la valeur de F_i exprimée par une fonction du vecteur d'état global X . Cette fonction de X permet ainsi, lorsqu'elle existe, de déterminer la dynamique du système de production. Plus précisément, nous montrons dans la section 3.2.4.1, qu'elle permet de déterminer la dynamique au sens de la définition 3.3.a. Une dynamique au sens de la définition 3.3.a est déterminée lorsque D , EU et E sont définis.

Avec l'usage d'un critère de type (b), un résultat similaire est obtenu section 3.2.4.2. Il est montré qu'une application qui exprime M_i en fonction de X peut déterminer la dynamique d'un système de production au sens de la définition 3.3.b. Une dynamique au sens de la définition 3.3.b est déterminée lorsque D' , $E'U'$ et E' sont définis.

Dans la section 3.2.4.3, des résultats supplémentaires sont recherchés. On montre qu'avec F_i fonction de X , il est possible d'exprimer la dynamique selon la définition 3.3.b. On montre

par contre qu'il n'est pas possible avec M_i fonction de X , d'exprimer la dynamique selon la définition 3.3.a.

La détermination de la dynamique par F_i ou M_i en fonction de X nous permettra dans le chapitre 5 de faire des comparaisons entre dynamiques uniquement en comparant ces fonctions.

3.2.4.1 Utilisation de F_i fonction de X pour un critère de type (a)

Dans cette section, on utilise la valeur des F_i exprimée en fonction du vecteur d'état global X pour obtenir les informations nécessaires à la détermination d'une dynamique selon la définition 3.3.a. Pour une ligne de production ayant une dynamique définie, l'ensemble E des états admissibles est connu et l'ensemble de tous les vecteurs X associés peut alors être trouvé. Nous allons maintenant définir cet ensemble de vecteurs d'état global admissibles.

Définition 3.4.a :

E_X est l'ensemble des vecteurs d'état global admissibles associés à E , l'ensemble des vecteurs d'états admissibles pour les critères de type (a). On a $E_X = \mathfrak{X}[E]$, (\mathfrak{X} étant définie en 3.2.a).

□

Cette définition caractérise E_X comme étant l'image de E par l'application \mathfrak{X} qui utilise les formules (3.2) et (3.3). Nous allons montrer que sous certaines conditions, il existe une bijection entre E_X et E , ces deux ensembles sont alors équipotents. A partir d'une fonction $f_i(X)$ nous allons définir une application \mathfrak{e} qui est l'inverse de l'application \mathfrak{X} .

Définition 3.5.a :

On définit l'application \mathfrak{e} par :

$$\begin{array}{ccc} \mathfrak{e} : E_X & \longrightarrow & E \\ X & \longmapsto & e \end{array}$$

avec

$$\begin{aligned} e_{D_N} &= -\min(X_N, 0) & e_{P_N} &= \max(X_N, 0) \\ e_{P_i} &= X_i - e_{F_{i+1}} & \text{pour } i &\in \{1; \dots; N-1\} \\ e_{F_i} &= f_i(X) & \text{pour } i &\in \{1; \dots; N\} \\ e_{C_i} &= \infty \end{aligned}$$

où f_i est une application définie de E_X sur \mathbb{N} pour i allant de 1 à N et vérifiant pour tout e de E :

$$e_{F_i} = f_i(\mathfrak{X}(e)) \tag{3.14}$$

□

Les composantes e_{D_N} et e_{P_N} sont déduites de (3.5). L'application $f_i(X)$ correspond au nombre de pièces contenues dans le processus de fabrication F_i sachant que l'état global est X , elle

vaut donc e_{F_i} pour tout état e de E tel que $\mathfrak{X}(e) = X$, ceci correspond à la formule (3.14). Il n'est pas possible de définir l'application \mathfrak{e} si f_i ne peut être déterminée à partir de cette formule. Comme cela est montré dans l'exemple de la remarque 3.2 qui suit, il existe certains cas où f_i n'existe pas.

Lemme 3.1.a :

Pour une ligne de production, dont la dynamique est déterminée selon la définition 3.3.a, si \mathfrak{e} existe conformément à la définition 3.5.a alors E_X et E sont équipotents et $E = \mathfrak{e}[E_X]$.

□

Lorsque \mathfrak{e} existe, $f_i(X)$ est définie et vérifie (3.14). Il est alors possible de montrer qu'il existe une bijection entre les états globaux et les autres états éléments de E .

Preuve :

On trouve $\mathfrak{X}[E] = E_X$ qui se déduit directement de la définition 3.4.a. Montrons que \mathfrak{X} est la fonction inverse de \mathfrak{e} . Il faut donc montrer que pour tout élément e de E on a $e = \mathfrak{e}(\mathfrak{X}(e))$.

- Avec (3.5) on trouve $e_D = -\min(0, e_{P_N} - e_D)$ et $e_{P_N} = -\min(e_D - e_{P_N}, 0)$ donc $e_D = \mathfrak{e}(\mathfrak{X}(e))_D$ et $e_{P_N} = \mathfrak{e}(\mathfrak{X}(e))_{P_N}$.
- Pour i allant de 1 à $N-1$ on trouve $\mathfrak{e}(\mathfrak{X}(e))_{P_i} = e_{P_i} + e_{F_{i+1}} - f_{i+1}(\mathfrak{X}(e))$. D'après les hypothèses du lemme $e_{F_{i+1}} = f_{i+1}(\mathfrak{X}(e))$ donc $\mathfrak{e}(\mathfrak{X}(e))_{P_i} = e_{P_i}$.
- De même on a $\mathfrak{e}(\mathfrak{X}(e))_{F_i} = f_i(\mathfrak{X}(e)) = e_{F_i}$ pour i allant de 1 à N .

Ainsi $\mathfrak{X}(\cdot) = \mathfrak{e}^{-1}(\cdot)$, E_X et E sont équipotents et $\mathfrak{e}[E_X] = E$.

□

Remarque 3.1 :

S'il existe une dynamique selon la définition 3.3.a alors lorsqu'un événement \emptyset , u_D , u_{F_1} ou u_{F_i} se produit, l'état global X élément de E_X varie respectivement de 0, -1_N , 1_1 et $1_i - 1_{i-1}$. Lorsqu'un événement se produit, la variation des variables d'état global X est donc connu sans qu'il soit nécessaire de connaître la dynamique du système. Le lemme 3.1.a permet de dire que, si \mathfrak{e} existe, alors pour un événement donné, les variations d'un vecteur d'état e de E se déduisent de celles des vecteurs d'état global X de E_X , car E_X et E sont équipotents. L'existence de \mathfrak{e} nécessite bien sûr l'existence de $f_i(X)$.

Remarque 3.2 :

Dans certains cas \mathfrak{e} n'existe pas et donc il n'y a pas toujours équipotence entre E_X et E même si la dynamique est déterminée selon la définition 3.3.a. La figure 3.8 montre un contre-exemple sur lequel on a défini une politique partielle où un couple (état, événement) fait correspondre un autre état. En partant d'un même état, suivant l'ordre d'arrivée des événements, u_{F_1} puis u_{F_2} (partie gauche de la figure 3.8) ou u_{F_2} puis u_{F_1} (partie droite de la

figure 3.8), on observe qu'il est possible d'atteindre un même état global avec $X_1 = 1$ et $X_2 = 2$ mais cependant les états sont différents puisque $P_1 = 0$ et $F_2 = 1$ sur la partie gauche de la figure 3.8 tandis qu'à droite $P_1 = 1$ et $F_2 = 0$. Les contre-exemples sont rares, celui de la figure 3.8 ne correspond d'ailleurs pas à une politique existante, il est présenté ici uniquement pour montrer que d'un point de vue théorique et mathématique il existe un contre-exemple.

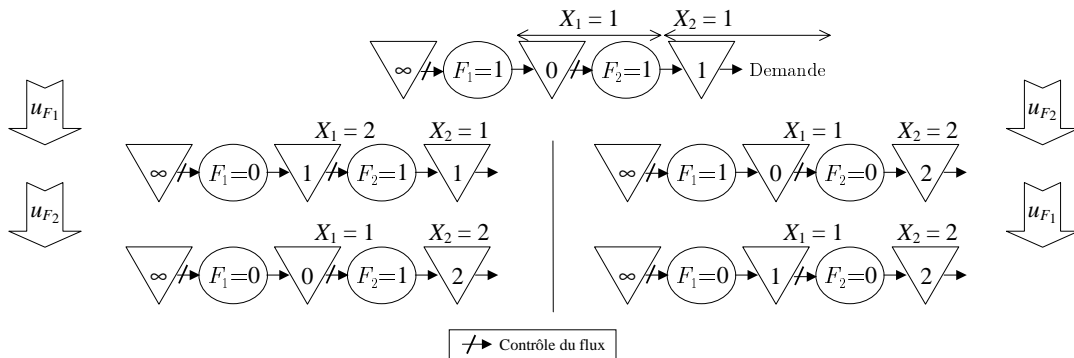


Figure 3.8 : Exemple de dynamique où pour deux états différents il y a un même état global

Les fonctions $f_i(X)$ peuvent suffire, lorsqu'elles existent, à exprimer la dynamique du système de production, ceci est montré par la proposition qui suit.

Proposition 3.1.a :

Dans une ligne de production où il existe une dynamique selon la définition 3.3.a, si \mathfrak{e} existe conformément à la définition 3.5.a, alors on peut caractériser cette dynamique à partir de \mathfrak{e} avec f_i et E_X . □

Preuve :

Le lemme 3.1.a s'applique et il montre qu'il y a correspondance entre un état dit global de E_X et un état de E utilisé dans la définition 3.3.a. A chaque élément $X \in E_X$ on associe l'élément $e = \mathfrak{e}(X)$, ce qui permet d'obtenir :

$$E = \mathfrak{e}[E_X]$$

Pour obtenir une dynamique conforme à la définition 3.3.a, les seuls événements possibles pour chacun des états sont déterminés par l'ensemble U_e de la définition 3.1.a. L'ensemble UE est défini par E et U_e :

$$EU = \{(e, u) \mid e \in E; u \in U_e\}$$

Pour savoir s'il est possible de déterminer de façon unique la dynamique il suffit de trouver une application D définie de EU vers E qui vérifie les propriétés (3.5), (3.6), (3.7), (3.8) et de montrer qu'il n'en existe pas d'autre.

Suite à la définition de E la propriété (3.5) est vérifiée car :

$$\min(e_D, e_{P_N}) = \min(-\min(X_N, 0), -\min(-X_N, 0)) = 0$$

L'application D doit vérifier les propriétés (3.6), (3.7) et (3.8). Ces propriétés permettent de déterminer le changement d'état global qui suit l'arrivée de tout événement. L'application D détermine le changement d'état élément de E qui suit l'arrivée de tout événement. Comme à tout état global de E_X correspond un unique état de E on trouve alors toutes les valeurs prises par D :

- Pour tout couple $(e, \emptyset) \in EU$, on a (3.6) c'est-à-dire $D(e, \emptyset) = e$.
- Pour tout couple $(e, u_D) \in EU$ en utilisant (3.7) et le Lemme 3.1.a on trouve :

$$D(e, u_D) = \mathfrak{e}(\mathfrak{X}(D(e, u_D))) = \mathfrak{e}(\mathfrak{X}(e) - 1_N)$$

- Pour les couples $(e, u_{F_i}) \in EU$ en utilisant (3.8) et le Lemme 3.1.a avec $1 < i \leq N$ on trouve :

$$D(e, u_{F_i}) = \mathfrak{e}(\mathfrak{X}(D(e, u_{F_i}))) = \mathfrak{e}(\mathfrak{X}(e) + 1_i - 1_{i-1}).$$

- De même pour les couples $(e, u_{F_1}) \in EU$ on obtient $D(e, u_{F_1}) = \mathfrak{e}(\mathfrak{X}(e) + 1_1)$.

L'application D est déterminée de façon unique. Tous les éléments nécessaires ont été trouvés de façon unique, donc on a bien déterminé la dynamique au sens de la définition 3.3.a. □

En général le fonctionnement d'un système de production est déterminé par une politique de gestion de production et par le paramétrage de cette politique ; pour chaque état on peut alors définir les événements pouvant se produire et le changement d'état qui est provoqué, ceci revient à déterminer la dynamique au sens de la définition 3.3.a. La proposition 3.1.a qui précède nous montre qu'il est possible de déterminer la dynamique au sens de la définition 3.3.a avec des fonctions f_i définies sur E_X correspondant aux valeurs prises par F_i . Pour définir la dynamique des politiques, avec un critère de type (a), il est donc suffisant d'exprimer la valeur des F_i en fonction de chaque état global X admissible (cela sera réalisé pour différentes politiques dans le chapitre 4).

Nous allons à présent étendre ces résultats au cas d'un système où les états significatifs dépendent d'un critère de type (b).

3.2.4.2 Utilisation de M_i fonction de X pour un critère de type (b)

Les systèmes de production considérés comportent des processus de fabrication composés d'un stock tampon infini suivi d'une station monoserveur, et les critères considérés sont de type (b). Ainsi il est nécessaire dans ce cas d'utiliser la notion de dynamique au sens de la définition 3.3.b.

Nous montrons dans cette section qu'il est possible, à partir d'une expression de M_i fonction de X , de déterminer la dynamique d'un système au sens de la définition 3.3.b. Pour atteindre cet objectif il suffit d'adapter au critère de type (b) les définitions, le lemme et la proposition de la précédente section.

Définition 3.4.b :

E'_X est l'ensemble des vecteurs d'état global admissibles associés à E' , l'ensemble des vecteurs d'états admissibles pour les critères de type (b). On a $E'_X = \mathfrak{X}'[E']$

□

Définition 3.5.b :

On définit l'application \mathfrak{e}' par :

$$\begin{array}{ccc} \mathfrak{e}' : E'_X & \longrightarrow & E' \\ X & \mapsto & e \end{array}$$

avec

$$\begin{array}{lll} e_{D_N} = -\min(X_N, 0) & e_{M_i} = m_i(X) & \\ e_{P_N} = -\min(-X_N, 0) & e_{P'_i} = X_i - f_{i+1}(X) & e_{C'_i} = 1 - m_i(X) \end{array}$$

où m_i est une application définie de E'_X sur \mathbb{N} pour i allant de 1 à N et vérifiant pour tout e de E' :

$$e_{M_i} = m_i(X'(e)) \tag{3.15}$$

□

Lemme 3.1.b :

Pour une ligne de production, dont la dynamique est déterminée selon la définition 3.3.b, si \mathfrak{e}' existe conformément à la définition 3.5.b alors E'_X et E' sont équipotents et $E' = \mathfrak{e}'[E'_X]$.

□

(Preuve en annexe B à la section B.I.)

Proposition 3.1.b :

Dans une ligne de production où il existe une dynamique selon la définition 3.3.b, si \mathfrak{e}' existe conformément à la définition 3.5.b, alors on peut caractériser cette dynamique à partir de \mathfrak{e}' avec m_i et E'_X .

□

(Preuve en annexe B à la section B.I.)

3.2.4.3 Relations entre les deux types de dynamique.

Dans cette section on met en évidence les différences et les liens qui existent entre les précédents résultats obtenus pour les deux types de dynamique.

On commence par montrer que E_X est égal à E'_X (lemme 3.2). Ensuite on s'aperçoit ensuite par un exemple que la connaissance d'une dynamique selon la définition 3.3.a ne permet pas

toujours d'en déduire une dynamique selon la définition 3.3.b. Toujours par un exemple on remarque qu'un état de E' peut correspondre à plusieurs états de E donc en connaissant une dynamique selon la définition 3.3.b, plusieurs dynamiques selon la définition 3.3.a sont possibles.

Le lemme 3.3 montre que E' est l'image de E par une application. On montre alors que si F_i fonction de X existe il est possible de trouver la dynamique selon la définition 3.3.b à partir d'une dynamique selon la définition 3.3.a (proposition 3.2). Il est même possible de trouver la dynamique selon la définition 3.3.b uniquement à partir de F_i fonction de X (corollaire 3.1).

Lemme 3.2 :

$$E_X = E'_X \quad \square$$

Preuve :

Les équations (3.2) et (3.3) sont vérifiées, cela signifie que pour tout état e de E il existe un état e' de E' tel que $\mathfrak{X}(e) = \mathfrak{X}'(e')$ et réciproquement pour tout état e' de E' il existe un état e de E tel que $\mathfrak{X}(e) = \mathfrak{X}'(e')$. Or pour tout $X \in E_X$ il existe un vecteur d'état $e \in E$ tel que $X = \mathfrak{X}(e)$. Il existe donc $e' \in E'$ tel que $\mathfrak{X}(e) = \mathfrak{X}'(e')$ ce qui montre que $X \in E'_X$ et ainsi on obtient $E_X \subset E'_X$.

De la même façon on montre aisément que $E'_X \subset E_X$, ceci démontre le lemme. \square

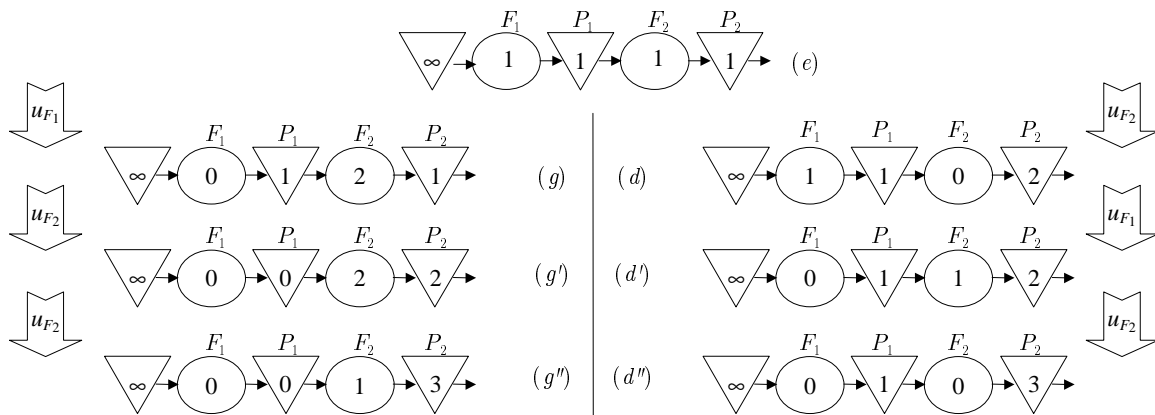


Figure 3.9 : Exemple où une dynamique peut exister au sens de la définition 3.3.a mais pas selon la définition 3.3.b

La figure 3.9 montre sur un exemple qu'il est possible de définir une fonction D pour exprimer la dynamique correspondante au sens de la définition 3.3.a. Nommons e l'état représentant la ligne du haut au centre de la figure 3.9, g, g', g'' les états représentant de haut en bas les trois lignes de gauche et d, d', d'' les états représentant de haut en bas les trois lignes de droite. Ces états sont élément de E et on a :

$$e = (e_{D_2}, e_{P_1}, e_{P_2}, e_{F_1}, e_{F_2}, e_{C_1}, e_{C_2}) = (0, 1, 1, 1, 1, \infty, \infty);$$

$$g = (0, 1, 1, 0, 2, \infty, \infty); d = (0, 1, 2, 1, 0, \infty, \infty);$$

$$g' = (0, 0, 2, 0, 2, \infty, \infty); d' = (0, 1, 2, 0, 1, \infty, \infty);$$

$$g'' = (0, 0, 3, 0, 1, \infty, \infty); d'' = (0, 1, 3, 0, 0, \infty, \infty);$$

On peut écrire $D(e, u_{F_1}) = g$, $D(g, u_{F_2}) = g'$, $D(g', u_{F_2}) = g''$, $D(e, u_{F_2}) = d$, $D(d, u_{F_1}) = d'$ et $D(d', u_{F_2}) = d''$.

Cette fonction D peut être utilisée pour définir la dynamique au sens de la définition 3.3.a. Par contre dans cet exemple on ne peut pas définir une fonction D' pour décrire la dynamique au sens de la définition 3.3.b car dans E' les états correspondant à e , g , d , g' , d' , g'' , d'' sont respectivement :

$$(e_{D_2}, e_{P'_1}, e_{P'_2}, e_{M_1}, e_{M_2}, e_{C'_1}, e_{C'_2}) = (0, 1, 1, 1, 1, 0, 0);$$

$$(0, 2, 1, 0, 1, 1, 0); (0, 1, 2, 1, 0, 0, 1);$$

$$(0, 1, 2, 0, 1, 1, 0); (0, 1, 2, 0, 1, 1, 0);$$

$$(0, 0, 3, 0, 1, 1, 0); (0, 1, 3, 0, 0, 1, 1);$$

Les états g' et d' correspondent au même vecteur $(0, 1, 2, 0, 1, 1, 0)$ de E' . lorsqu'une pièce sort du processus de fabrication 2 et que l'état est $(0, 1, 2, 0, 1, 1, 0)$ on peut alors aller vers deux états différents. Ainsi $D'((0, 1, 2, 0, 1, 1, 0), u_{F_2})$ n'est pas définie car chaque valeur $(0, 0, 3, 0, 1, 1, 0)$ et $(0, 1, 3, 0, 0, 1, 1)$ de E' serait possible. Dans le cas général il n'est donc pas toujours possible, à partir d'une dynamique au sens de la définition 3.3.a, d'en trouver une au sens de la définition 3.3.b.

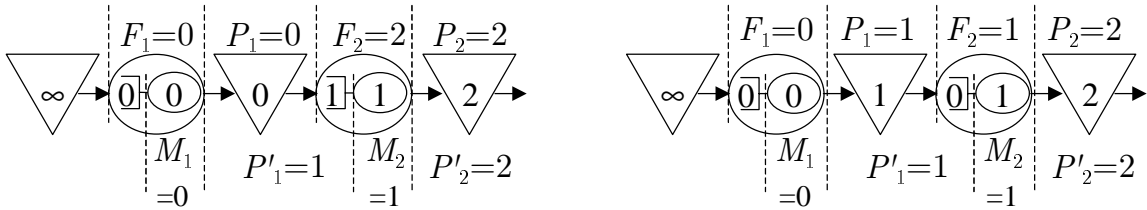


Figure 3.10 : Exemple de deux états différents pour la définition 3.3.a mais identiques pour la définition 3.3.b

On montre aussi qu'il n'est pas toujours possible de trouver une dynamique au sens de la définition 3.3.a, lorsque l'on ne connaît que celle donnée au sens de la définition 3.3.b. La figure 3.10 montre un exemple où pour un état de E' il existe plusieurs états de E possibles, ceci empêche la détermination de la dynamique au sens de la définition 3.3.a. Dès qu'un stock P'_i est non vide il est impossible de savoir si son contenu est dans la file d'attente tampon de F_{i+1} ou dans le stock P_i . Donc si l'on ne possède que les seules informations données par une dynamique au sens de la définition 3.3.b, plusieurs dynamiques au sens de la définition 3.3.a peuvent alors correspondre.

En général E et E' ne sont pas équipotents. Cependant lorsque l'on connaît la valeur de P_i et de F_i il est facile (puisque F_i est composé d'une file d'attente et d'un monoserveur) d'en déduire la valeur de P'_i et M_i , on parvient ainsi à trouver une application surjective définie de E vers E' :

Définition 3.6 :

L'application V est définie par :

$$V : E \longrightarrow V[E]$$

$$e \mapsto e' = (e'_{D'}, e'_{P'_1}, \dots, e'_{P'_N}, e'_{M_1}, \dots, e'_{M_N}, e'_{C'_1}, \dots, e'_{C'_N})$$

$$\text{où } e'_D = e_D, e'_{P'_N} = e_{P_N}, e'_{M_i} = \min(1, e_{F_i}), e'_{P'_i} = e_{P_i} + e_{F_{i+1}} - e'_{M_i}, e'_{C'_i} = 1 - e'_{M_i}$$

□

Lemme 3.3 :

Si E est connu alors $E' = V[E]$.

□

Preuve :

Les propriétés (3.2), (3.3) sont vérifiées donc pour tout élément e de E il existe un élément e' de E' vérifiant $\mathfrak{X}(e) = \mathfrak{X}'(e')$ et réciproquement pour tout élément e' de E' il existe un élément e de E vérifiant $\mathfrak{X}(e) = \mathfrak{X}'(e')$. Avec la propriété (3.1) on montre que $e'_{D_N} = e_{D_N}$ et $e'_{P'_N} = e_{P_N}$. La propriété (3.4) permet de montrer que pour $i \in \{1, \dots, N\}$ on a $e'_{M_i} = \min(1, e_{F_i})$ et $e'_{C'_i} = 1 - e'_{M_i}$. Avec (3.2), et (3.3) on obtient pour $i \in \{1, \dots, N-1\}$, $e'_{P'_i} = e_{P_i} + e_{F_{i+1}} - e'_{M_{i+1}}$. Ainsi pour chaque élément e de E on a $V(e) \in E'$ et pour chaque élément e' de E' il existe un élément e de E tel que $e' = V(e)$. On a bien $E' = V[E]$.

□

Pour chaque état de E l'application V permet d'obtenir l'état qui lui correspond dans E' . On peut aussi trouver pour chaque couple état événement de EU le couple dans $E'U'$ qui lui correspond. L'application qui suit va nous permettre d'obtenir de telles correspondances entre EU et $E'U'$.

Définition 3.7 :

L'application W est définie par :

$$W : EU \longrightarrow W[EU]$$

$$(e, u) \mapsto (V(e), u_{MC'_i}) \text{ si } u = u_{F_i} \text{ et } (V(e), u) \text{ sinon}$$

□

Proposition 3.2 :

Soit une ligne de production où l'application D qui est définie de E vers EU détermine la dynamique au sens de la définition 3.3.a, si \mathfrak{e} existe conformément à la définition 3.5.a alors on en déduit qu'il existe une unique dynamique au sens de la définition 3.3.b. Elle est déterminée par :

$$D' = V \circ D \circ W^{-1} \text{ définie de } E' = V[E] \text{ vers } E'U' = W[EU]$$

□

Preuve :

Montrons que W est bijective. Par définition V et W sont surjectives. Supposons qu'il existe $W(e, u) = W(f, v)$ alors si $u = u_{F_i}$ on a $(V(e), u_{MC'_i}) = (V(f), u_{MC'_i})$ et $v = u_{F_i} = u$, sinon on a $(V(e), u) = (V(f), v)$, ainsi on a toujours $u = v$ et $V(e) = V(f)$.

On peut montrer facilement en utilisant $V(e) = V(f)$ que e et f respectent $e_D = f_D$, $e_{P_N} = f_{P_N}$, $e_{P_i} + e_{F_{i+1}} = f_{P_i} + f_{F_{i+1}}$ et $\min(1, e_{F_i}) = \min(1, f_{F_i})$. On remarque alors que $\mathfrak{X}(e) = \mathfrak{X}(f)$.

L'existence de \mathfrak{e} implique l'existence des fonctions f_i définies de E_X sur \mathbb{N} et vérifiant $e_{F_i} = f_i(\mathfrak{X}(e))$ pour tout e de E . Ainsi pour $i \in \{1, \dots, N\}$ on trouve que $e_{F_i} = f_i(\mathfrak{X}(e)) = f_i(\mathfrak{X}(f)) = f_{F_i}$. On a $e_{P_i} + e_{F_{i+1}} = f_{P_i} + f_{F_{i+1}}$ donc $e_{P_i} = f_{P_i}$. Tout ceci montre que les applications V et W sont donc bijectives, et que D' est bien défini.

On peut vérifier facilement que toutes les propriétés de la définition 3.3.b sont vérifiées par l'application D' . On vient de montrer que la proposition exhibait une dynamique au sens de la définition 3.3.b, montrons qu'il n'en existe pas d'autre.

En utilisant le lemme 3.3 on trouve $E' = V[E]$. Pour toute application D' vérifiant les propriétés de la définition 3.3.b et pour tout e de E et $e' = V(e)$ on obtient ce qui suit :

- L'ensemble U_e est décrit dans la définition 3.1.a par

$$U_e = \{\emptyset, u_D\} \cup \{u_{F_i} \mid e_{F_i} > 0, i \in \{1, \dots, N\}\}$$

et U'_e est décrit dans la définition 3.1.b par

$$U'_e = \{\emptyset, u_D\} \cup \{u_{MC'_i} \mid e_{M_i} > 0, i \in \{1, \dots, N\}\}.$$

Avec (3.4) on obtient

$$(V(e))_{M_i} = e'_{M_i} \neq 0 \Leftrightarrow e_{F_i} \neq 0$$

et on trouve alors

$$E'U' = \{(e', u') \mid e' \in E', u' \in U'_e\} = \{W(e, u) \mid e \in E, u \in U_e\} = W[EU].$$

- Avec (3.11) on a $D'(e', \emptyset) = e' = V \circ D \circ W^{-1}(e, \emptyset)$.
- Avec (3.12) et (3.13) on obtient:

$$\mathfrak{X}'(D'(e', u_D)) = \mathfrak{X}'(e') - 1_N$$

$$\mathfrak{X}'(D'(e', u_{MC'_1})) = \mathfrak{X}'(e') + 1_1$$

$$\forall i \in \{2, \dots, N\}, \mathfrak{X}'(D'(e', u_{MC'_i})) = \mathfrak{X}'(e') - 1_{i-1} + 1_i$$

de même avec (3.7) et (3.8) on a

$$\mathfrak{X}(D(e, u_D)) = \mathfrak{X}(e) - 1_N$$

$$\mathfrak{X}(D(e, u_{F_1})) = \mathfrak{X}(e) + 1_1.$$

$$\forall i \in \{2, \dots, N\}, \mathfrak{X}(D(e, u_{F_i})) = \mathfrak{X}(e) - 1_{i-1} + 1_i$$

On montre avec (3.2) et (3.3) que $\mathfrak{X}'(e') = \mathfrak{X}'(V(e)) = \mathfrak{X}'(e)$ et on obtient alors ce qui suit :

$$\mathfrak{X}'(D'(e', u_D)) = \mathfrak{X}(D(e, u_D)) = \mathfrak{X}(D \circ W^{-1}(e', u_D)) \text{ et}$$

$$\forall i \in \{1, \dots, N\}, \mathfrak{X}'(D'(e', u_{MC'_i})) = \mathfrak{X}(D(e, u_{F_i})) = \mathfrak{X}(D \circ W^{-1}(e', u_{MC'_i})).$$

Supposons que nous ayons $\mathfrak{X}'(f') = \mathfrak{X}(f)$ alors avec (3.1), (3.2), (3.3) et (3.4) on montre que $f' = V(f)$. Ceci permet de modifier le précédent résultat, on trouve :

$$D'(e', u_D) = V \circ D \circ W^{-1}(e', u_D) \text{ et}$$

$$\forall i \in \{1, \dots, N\}, D'(e', u_{MC'_i}) = V \circ D \circ W^{-1}(e', u_{MC'_i}).$$

Il n'y a qu'une façon de définir D' , le résultat qui est alors obtenu est égal à $V \circ D \circ W^{-1}$, la proposition est démontrée. \square

Dans la proposition 3.2 l'existence de \mathfrak{E} est essentielle car, par exemple, si \mathfrak{E} existe on ne peut pas avoir le cas présenté sur la figure 3.9. Un résultat similaire à cette proposition, qui permettrait de définir D à partir de D' , ne peut pas être possible car même si \mathfrak{E}' existe le contre-exemple de la figure 3.10 reste valable.

La proposition 3.2 nécessite la connaissance de l'application D qui détermine la dynamique au sens de la définition 3.3.a et des fonctions f_i définies de E_X sur \mathbb{N} . Le corollaire qui suit atteint les mêmes objectifs sans qu'il soit nécessaire de connaître D . Il sera suffisant de savoir que la dynamique existe selon la définition 3.3.a.

Corollaire 3.1 :

Pour une ligne de production, dont la dynamique existe selon la définition 3.3.a, si \mathfrak{E} existe conformément à la définition 3.5.a. Si (3.1), (3.2), (3.3) et (3.4) sont vérifiées alors on peut déterminer avec f_i et E_X la dynamique de cette ligne de production au sens de la définition 3.3.b.

\square

Preuve :

On utilise la proposition 3.2 pour montrer qu'il existe une dynamique au sens de la définition 3.3.b. Le lemme 3.2 permet de montrer que $E'_X = E_X$. En utilisant la proposition 3.1.b et (3.4) on montre qu'il est possible de déterminer la dynamique de cette ligne de production au sens de la définition 3.3.b avec seulement f_i et E'_X .

□

3.2.5 Conclusions pour une ligne avec stations monoserveur précédées de stocks tampon

Pour un système dont le processus de fabrication est composé d'un stock tampon suivi d'une station monoserveur, il est possible de déterminer la dynamique du système uniquement par connaissance de F_i en fonction de l'état global. Ceci est possible pour un critère de type (a) (voir proposition 3.1.a) et pour un critère de type (b) (voir corollaire 3.1). Pour un critère de type (b) il est aussi possible de définir la dynamique de la ligne avec les valeurs de M_i en fonction de l'état global (voir proposition 3.1.b). La connaissance de la dynamique au sens de la définition 3.3.a implique celle au sens de la définition 3.3.b si F_i est connu en fonction de l'état global (voir proposition 3.2) mais le contraire n'est pas possible (voir figure 3.10). Dans le cas général lorsque F_i n'est pas connu en fonction de l'état global, une dynamique au sens de la définition 3.3.a peut exclure la possibilité d'en avoir une au sens de la définition 3.3.b (voir la figure 3.9).

L'étude de la dynamique d'un système peut être simplifiée par l'existence et la connaissance de fonctions déterminant les valeurs de F_i à partir du seul état global. Dans ce cas la dimension des vecteurs d'état est diminuée de moitié au moins sans qu'il y ait de perte d'information (voir lemmes 3.1.a et 3.1.b). La connaissance de F_i en fonction du seul état global suffit alors à exprimer la dynamique du système. La section suivante a pour objectif d'étendre ces résultats à des processus de fabrication quelconques.

3.3 Dynamique et formulation des politiques avec généralisation du type de processus de fabrication utilisé dans la ligne.

Pour le cas d'un processus de fabrication général (figure 3.11), une difficulté supplémentaire apparaît. Comme précédemment F_i (ou M_i) peut ne pas être défini en fonction de X à cause de la politique de gestion de production, cela est visible figure 3.9. Nous montrons dans cette section que F_i (ou M_i) peut aussi ne pas être défini à cause du processus de fabrication lui-même. En effet, la modification de l'état interne du processus de fabrication peut créer des blocages sur l'entrée de F_i (ou M_i) et modifier ainsi la valeur F_i (ou M_i). Puisque ces états

internes ne dépendent pas de X alors F_i (ou M_i) ne peut plus dépendre seulement de X . Ainsi dans certains cas, notre expression de la dynamique à partir de F_i (ou M_i) fonction de X , ne peut pas être utilisée directement. Nous montrons cependant, avec un exemple, qu'il existe une solution consistant à effectuer quelques petites modifications du vecteur d'état X .

Contrairement à la section 3.2, il est possible qu'un processus de fabrication ne possède pas de stock tampon, alors plutôt que de considérer que $P'_i = P_i$ et $M_i = F_i$ on considère que P'_i et M_i n'existent plus car leur définition respective n'a de sens qu'en présence d'un stock tampon infini. Ainsi seuls P_i et F_i sont définis (figure 3.11-b). Dans ce cas, l'espace d'état ne peut être défini que pour un critère de type (a). Dans les sections 3.3.2 et 3.3.3.2, on traite le cas d'un critère de type (b), mais on suppose alors que les processus de fabrication possèdent tous des stocks tampon.

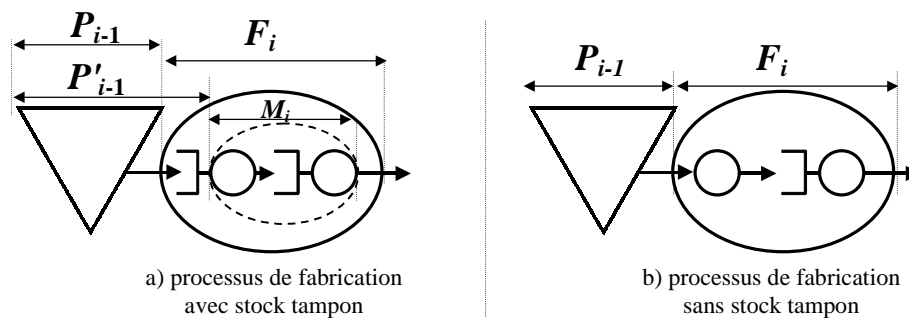


Figure 3.11 : Exemple de processus de fabrication quelconques avec ou sans stock tampon infini.

3.3.1 Définition de la notion de dynamique pour un critère de type (a)

Nous allons maintenant donner en fonction du critère de type (a) utilisé, une définition de la dynamique d'une ligne de production pour des processus de fabrication quelconques. Les processus de fabrication sont quelconques mais doivent cependant fonctionner à flux poussé. Ainsi lorsqu'une pièce rentre dans un processus de fabrication son service s'effectue indépendamment de tout ce qui peut se produire à l'extérieur de celui-ci. La capacité d'accueil pour un processus avec stock tampon est infinie alors que pour un processus de fabrication dans le cas général ce n'est pas toujours le cas. Ainsi un système peut avoir des variations de ses capacités d'accueil. Ces variations sont associées à l'utilisation de nouvelles classes d'événements dans la définition de la dynamique qui va suivre.

Comme précédemment pour définir la dynamique des lignes de fabrication nous utiliserons une application qui à chaque couple (état initial, événement) va associer un état final. On note EU l'ensemble des couples composés d'un état de E et d'un événement. Pour chaque état e de E on note U_e tous les événements discrets pouvant se produire. A chaque instant il n'y a qu'un événement possible, cependant dans le cas général certains événement peuvent correspondre à l'arrivée de plusieurs événements de base. On peut avoir une demande de q produits finis, une

sortie de j pièces d'un processus de fabrication, une augmentation de k de la capacité d'accueil ou encore un mélange des deux derniers événements.

Suivant le type de système de production étudié, l'ensemble U_e n'autorisera que certains événements. Pour les systèmes de production considérés, l'arrivée des demandes ne dépendra pas de l'état du système. On considère que, dans un processus de fabrication, l'instant où les pièces sortent de ce processus de fabrication ne dépend que de l'état initial et des instants où des pièces y sont entrées.

Pour un système de production où les demandes arrivent une à une et où les pièces sortent une à une des processus de fabrication, l'ensemble U_e ne devra contenir que les événements :

- ◆ Arrivée d'une demande.

Et pour chaque maille i :

- ◆ Sortie d'une pièce de F_i sans augmentation de la capacité d'accueil.
- ◆ Sortie d'une pièce de F_i avec augmentation simultanée de un de la capacité d'accueil.
- ◆ Augmentation de la capacité d'accueil de F_i avec aucune sortie de pièce de F_i .

Par contre pour un système de production par lots, on se limitera à des événements correspondants à la taille des lots. Si F_i produit des lots de taille Q_i alors l'ensemble U_e ne devra contenir que les événements :

- ◆ Arrivée d'une demande.

Et pour chaque maille i :

- ◆ Sortie de Q_i pièces de F_i sans augmentation de la capacité d'accueil.
- ◆ Sortie de Q_i pièces de F_i avec augmentation simultanée de k de la capacité d'accueil.
- ◆ Augmentation de k de la capacité d'accueil de F_i avec aucune sortie de pièce de F_i .

Les valeurs possibles de k sont déterminées en fonction de la nature du processus de fabrication.

Dans notre définition de la dynamique d'un système de production, nous utilisons E , EU et une application D définie de E vers EU .

L'ensemble E comprend des vecteurs d'état dont les $2N+1$ premières composantes sont entières et les N dernières qui représentent C_i sont entières ou infinies. Cet ensemble est inclus dans $\mathbb{N}^{1+2N} \times (\mathbb{N} \cup \{\infty\})^N$ car suivant la politique ou le type de ligne de production, certains états n'existent pas.

L'ensemble U_e des événements va dépendre du type de ligne utilisée, il est défini comme étant inclus dans un ensemble plus vaste d'événements.

Pour s'assurer que le système étudié puisse correspondre réellement à une ligne de production, plusieurs propriétés sont utilisées. Ceci est fait en annexe B, dans la section B.II, par une généralisation de la définition 3.3.a définissant la dynamique pour un critère de type (a). Cette généralisation porte sur le type de processus de fabrication utilisé.

Cette définition permet de voir si deux lignes de production ont une dynamique identique. Pour que deux dynamiques soient identiques il suffit (si la définition B.1.a de la section B.II de l'annexe B est utilisée) que les deux systèmes soient associés à une même application D définie sur un même ensemble EU et vers un même ensemble E .

3.3.2 Définition de la notion de dynamique pour un critère de type (b)

On peut de la même façon définir une généralisation de la dynamique pour un critère de type (b). Cette généralisation porte sur le type de processus de fabrication utilisé, elle est détaillée en annexe B à la section B.III. Rappelons que pour un critère de type (b) on ne considère que des cas où il y a un stock tampon en début de chaque processus de fabrication.

3.3.3 Détermination de la dynamique par F_i et C_i en fonction de X , ou par M_i et C'_i en fonction de X

Nous déterminons ici, comme à la section 3.2.4 mais pour des processus de fabrication plus généraux, la dynamique d'un système de production à partir de F_i et C_i en fonction de X lorsque le critère est de type (a) et à partir de M_i et C'_i en fonction de X lorsque le critère est de type (b). A la fin de cette section nous examinons les relations qu'il y a entre une dynamique définie pour un critère de type (a) et une dynamique définie pour un critère de type (b).

3.3.3.1 Détermination de la dynamique par F_i et C_i en fonction de X pour un critère de type (a).

Nous avons montré, pour une ligne de production composée de processus de fabrication fonctionnant à flux poussé, qu'il est possible de déterminer la dynamique au sens de la définition B.1.a en sachant que F_i et C_i sont égales à des applications définies de E_X vers E .

Un premier résultat consiste uniquement à montrer que de telles applications rendent E_X et E équipotents. Nous allons d'abord redéfinir l'application ϵ qui a déjà été précédemment définie pour un cas simple à la section 3.2.

Définition 3.8.a :

On définit l'application ϵ par :

$$\begin{array}{ccc} \epsilon : E_X & \longrightarrow & E \\ X & \longmapsto & e \end{array}$$

avec

$$\begin{array}{lll} e_{D_N} = -\min(X_N, 0) & e_{F_i} = f_i(X) & \\ e_{P_N} = -\min(-X_N, 0) & e_{P_i} = X_i - f_{i+1}(X) & e_{C_i} = c_i(X) \end{array}$$

où f_i et c_i sont des applications définies respectivement de E_X sur \mathbb{N} et de E_X sur $\mathbb{N} \cup \{\infty\}$ pour i allant de 1 à N et vérifiant pour tout e de E :

$$e_{F_i} = f_i(\mathfrak{X}(e)) \quad \text{et} \quad e_{C_i} = c_i(\mathfrak{X}(e)) \quad (3.16)$$

□

Pour rester cohérent avec la définition 3.5.a, il suffit de remarquer que pour un système avec stock tampon, on a $c_i(X) = \infty$.

Lemme 3.4.a :

Pour une ligne de production, dont la dynamique est déterminée selon la définition B.1.a, si \mathfrak{e} existe conformément à la définition 3.8.a, alors E_X et E sont équipotents et $E = \mathfrak{e}[E_X]$.

□

(Voir preuve en annexe B à la section B.IV.)

En posant $c_i(X) = \infty$ on retrouve le lemme 3.1.a. On peut remarquer que les hypothèses du lemme 3.4.a (c'est-à-dire l'existence de \mathfrak{e} principalement) ne peuvent pas être vérifiées sans que l'ensemble U_e ne vérifie les deux propriétés du lemme B.1.a se trouvant dans l'annexe B.

La première propriété indique que, dans un état e , pour tout événement de U_e impliquant la sortie de j pièces d'un processus de fabrication i , il n'y a qu'une seule augmentation de k unités de la capacité d'accueil possible simultanément. Cette augmentation de k unités de la capacité d'accueil est donc fonction de l'état e et du nombre j de pièces qui sortent du processus de fabrication i .

La deuxième propriété indique que tout événement créant une augmentation de la capacité d'accueil dans un processus de fabrication i , implique également une sortie simultanée d'une ou plusieurs pièces de ce processus de fabrication i . Cette dernière propriété peut s'expliquer ainsi :

Si \mathfrak{e} existe alors C_i et F_i sont fonctions de X . Une augmentation de k de la capacité d'accueil dans F_i sans sortie de pièce implique une augmentation de k de $C_i + F_i$ avec X ne variant pas. On a $C_i + F_i$ fonction de X donc k ne peut être que nul, il n'y a donc pas d'augmentation de la capacité d'accueil sans sortie de pièces.

La partie gauche de la figure 3.12 montre une maille où le processus de fabrication est composé de deux stations monoserveur en série séparées par un stock tampon. Un événement (u_{C_i}) d'augmentation de la capacité d'accueil peut se produire chaque fois qu'un produit sort de la première station. On remarque alors comme cela est montré sur la partie droite de la figure 3.12 que C_i et F_i peuvent varier sans que l'on ait aucune variation de X . Ces quantités ne sont donc pas fonctions du vecteur X et \mathfrak{e} n'existe pas dans ce cas.

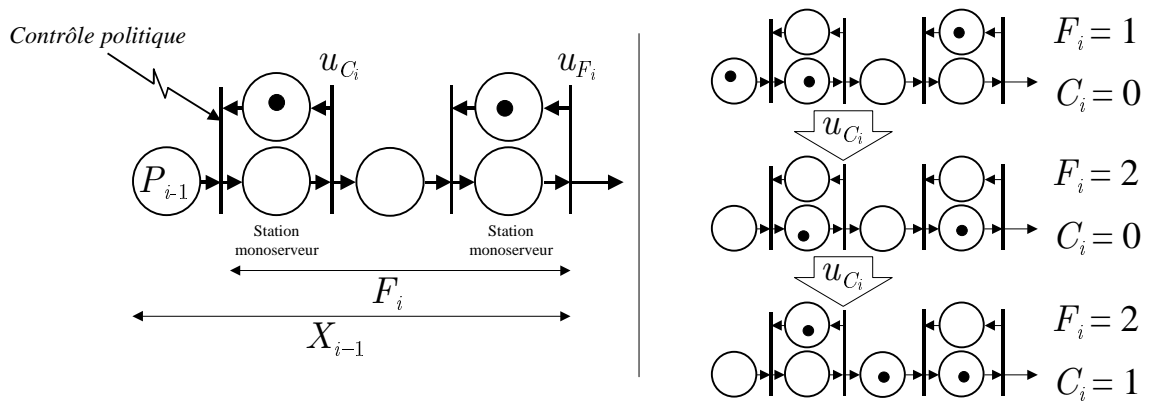


Figure 3.12 : Processus de fabrication composé de deux stations monoserveur en série.

Pour le cas de cette figure 3.12 les hypothèses du lemme 3.4.a ne peuvent donc pas être vérifiées. Cependant il est possible de contourner la difficulté.

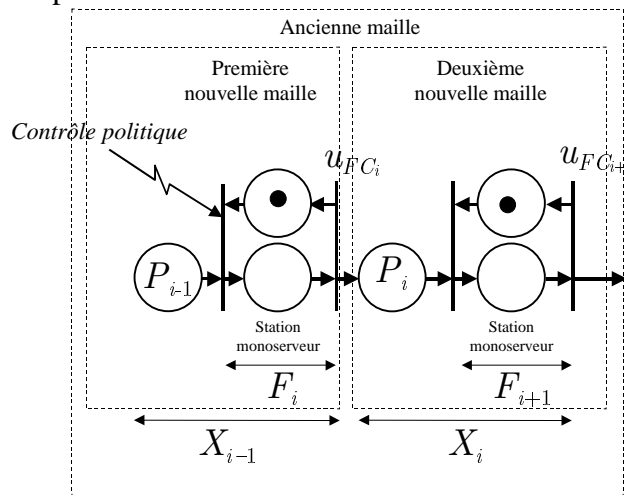


Figure 3.13 : Processus de fabrication composé de deux stations monoserveur en série .

La solution consiste à faire un découpage entre les deux stations et à dédoubler les mailles pour n'avoir qu'une station monoserveur par maille (voir figure 3.13). Le stock tampon situé entre les deux stations monoserveur joue le rôle de stock pour produits intermédiaires. Ainsi, avec les deux nouvelles mailles obtenues on peut vérifier les hypothèses du lemme 3.4.a et définir des fonctions $F_i(X)$, $F_{i+1}(X)$, $C_i(X)$ et $C_{i+1}(X)$.

Nous montrons à présent qu'il est possible de déterminer la dynamique au sens de la définition B.1.a en connaissant U_e , E_X et les applications f_i et c_i utilisées dans la définition de \mathfrak{e} .

Proposition 3.3.a :

Dans une ligne de production où il existe une dynamique selon la définition B.1.a, si \mathfrak{e} existe conformément à la définition 3.8.a, alors on peut caractériser cette dynamique à partir de \mathfrak{e} avec f_i , c_i , E_X et U_e .

□

(La preuve est donnée dans l'annexe B à la section B.IV)

Corollaire 3.2.a :

Si pour une ligne de production la valeur F_i de chaque étage i est fonction de X , si les demandes arrivent une à une et si un des points suivants est vérifié pour chaque maille i , alors on peut définir la dynamique de cette ligne de production selon la définition B.1.a.

- Un stock tampon infini est présent en début de processus de fabrication quelconque avec sortie des pièces par lots de taille Q_i .
- Un serveur infini est l'unique élément du processus de fabrication avec sortie des pièces par lots de taille Q_i .
- Le processus de fabrication n'a pas de file d'attente tampon, il est composé de n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i .

□

(La preuve est donnée dans l'annexe B à la section B.IV)

Pour des pièces servies une à une il suffit de fixer la taille des lots à un. Le corollaire 3.2.a montre que dans plusieurs cas avec un critère de type (a), la connaissance du nombre de pièces présentes dans les processus de fabrication en fonction de l'état global suffit à déterminer la dynamique du système. Pour les cas qui ne sont pas traités par le corollaire 3.2.a on peut utiliser la proposition 3.3.a qui permet de déterminer la dynamique d'une ligne de production relativement à un critère de type (a), à partir d'une application ϵ et de l'ensemble des événements U_e possibles pour chaque état e .

En utilisant un critère de type (b), des résultats similaires sont obtenus, la section suivante traite ce cas.

3.3.3.2 Détermination de la dynamique par M_i et C'_i en fonction de X pour un critère de type (b).

Dans cette section on utilise la généralisation de la dynamique pour critère de type (b) qui est définie section 3.3.2. La présence d'un stock tampon est donc nécessaire en début de chaque processus de fabrication. Dans la section 3.3.3.1, pour le critère de type (a) on a la définition 3.8.a, les lemmes 3.4.a et B.1.a, la proposition 3.3.a et le corollaire 3.2.a. De même dans cette section 3.3.3.2 qui a été déplacée en annexe B à la section B.V on a les mêmes éléments pour un critère de type (b), c'est-à-dire une définition 3.8.b, des lemmes 3.4.b et B.1.b, une proposition 3.3.b et un corollaire 3.2.b

(Voir en annexe B la section B.V pour plus de détails)

3.3.3.3 Relations entre les deux types de dynamique pour des processus de fabrication avec stock tampon et multiserveur.

Ici des résultats similaires à ceux de la section 3.2.4.3 ont été obtenus. Cependant au lieu de considérer des processus de fabrication constitués d'un stock tampon suivi d'un monoserveur

on utilise des processus de fabrication constitués d'un stock tampon suivi de multiserveur produisant par lots.

(Voir en annexe B la section B.VI pour plus de détails)

3.3.4 Conclusions pour des systèmes avec processus de fabrication quelconques à flux poussé

On s'est intéressé à des lignes de production dont les processus de fabrication étaient les plus généraux possible. Ces processus de fabrication ont cependant un flux sortant qui ne dépend que du flux de produits poussés à leur entrée. Par ailleurs, pour étudier les cas où un critère de type (b) serait utilisé il est bien entendu nécessaire de supposer que les processus de fabrication comportent un stock tampon.

A chaque type de processus de fabrication sont associés un certain nombre d'événements possibles en fonction de l'état. Ces événements sont définis par U_e et ils ne dépendent que de la nature des processus de fabrication. Précédemment, dans le cas de processus de fabrication avec stock tampon suivi d'un monoserveur, U_e a été défini par la définition 3.1.a. Dans cette section 3.3, U_e est défini en fonction des processus de fabrication utilisés. Dans la définition B.2.a les événements des ensembles U_e sont définis pour une ligne de multiserveur avec lot de taille Q_i , ils sont fonctions de la taille des lots. Pour un critère de type (b) des remarques similaires peuvent être faites avec U'_e .

Lorsqu'il est possible de connaître F_i et C_i en fonction de l'état global, il est alors possible de déterminer la dynamique au sens de la définition B.1.a uniquement en utilisant U_e , E_X et les F_i et C_i fonctions de l'état global (voir proposition 3.3.a). De même s'il est possible de connaître M_i et C'_i en fonction de l'état global, alors on peut déterminer la dynamique au sens de la définition B.1.b en utilisant U'_e , E'_X et chaque M_i et C'_i fonction de l'état global (voir proposition 3.3.b).

Dans le cas de processus de fabrication avec stock tampon suivi d'un multiserveur produisant par lots, nous avons mis en évidence qu'il existait quelques liens entre une dynamique déterminée au sens de la définition B.1.a et une au sens de la définition B.1.b. La connaissance de la dynamique au sens de la définition B.1.a implique celle au sens de la définition B.1.b si F_i est connu en fonction de l'état global (voir proposition B.1). Il est même suffisant de savoir que la dynamique au sens de la définition B.1.a existe pour déterminer la dynamique du système au sens de la définition B.1.b en utilisant F_i en fonction de l'état global (voir corollaire B.1).

Le lemme 3.4.a montre qu'il est possible de connaître un état e (élément de E) du système à partir de l'état global X lorsque que l'application ϵ de la définition 3.8.a existe. On utilise alors les valeurs de F_i et C_i fonctions de X qui sont nécessaires à l'existence de ϵ , pour obtenir e en fonction de X . De façon similaire il est possible de déterminer e élément de E' à

partir de l'état global X et des valeurs de M_i et C'_i obtenues en fonction de X . La possibilité de connaître F_i et C_i (respectivement M_i et C'_i) en fonction de l'état global X , impose aux ensembles U_e (respectivement U'_e) qu'ils respectent les propriétés du lemme B.1.a (respectivement B.1.b). Ces propriétés déterminent les augmentations possibles de la capacité d'accueil à l'intérieur des processus de fabrication pour qu'il soit possible à partir des variations de l'état global X de détecter l'occurrence de tout événement de U_e (ou de U'_e). Dans la figure 3.12, ces propriétés ne sont pas vérifiées. Ainsi pour ce cas, le lemme 3.4.a et la proposition 3.3.a ne peuvent pas s'appliquer.

Nous avons montré que pour de nombreux types de processus de fabrication il était possible de décrire la dynamique du système en utilisant les valeurs de F_i en fonction de l'état global. Dans le chapitre 4 nous allons montrer comment trouver ces valeurs de F_i en fonction de l'état global.

Calcul des fonctions utilisées dans la formulation unifiée de la dynamique des politiques à flux tiré

Dans le chapitre précédent il a été montré que le calcul de F_i en fonction de l'état global permettait de déterminer la dynamique de certains systèmes de production. Dans ce chapitre, nous effectuons ce calcul qui permettra dans les chapitres suivants de trouver des identités de fonctionnement entre politiques ou de calculer une fonction coût à partir de $F_i(X)$.

Plusieurs méthodes, dont une utilisant les invariants [BOL 1999.b], permettent d'obtenir la valeur de F_i en fonction X ; cependant nous pensons que la méthode la plus efficace consiste à utiliser l'algèbre (min,+). C'est d'ailleurs avec cette méthode qu'il nous a été possible de généraliser nos résultats.

Dans un premier temps, pour simplifier, nous ne considérons que les systèmes de production dont les processus de fabrication comportent un stock tampon suivi d'une station monoserveur. Ainsi pour retrouver la dynamique du système de production il suffira d'utiliser la proposition 3.1.a ou le corollaire 3.1 page 52. Le calcul de F_i en fonction de l'état global est alors réalisé pour les politiques kanban, basestock, kanban généralisé et kanban étendu.

Il est intéressant de remarquer que, pour les différentes politiques de gestion de production étudiées, les formules donnant la valeur de F_i en fonction de X sont très peu différentes les unes des autres. Seuls quelques paramètres changent dans ces formules. Pour une certaine catégorie de politiques de gestion de production modélisées par réseau de files d'attente, nous avons pu donner un algorithme qui calcule ces paramètres contenus dans les formules exprimant $F_i(X)$. Cet algorithme, qui nécessite la construction d'un graphe, trouve, par la recherche de quelques plus courts chemins, la valeur de ces paramètres.

Par la suite, nous considérons aussi des systèmes avec d'autres types de processus de fabrication. Le cas de quelques systèmes de production par lots puis avec station d'assemblage sont aussi étudiés.

4.1 Rappels sur l'utilisation de l'algèbre des dioïdes appliquée aux graphes d'événements

La modélisation du comportement des graphes d'événements temporisés est une des utilisations importantes de l'algèbre des dioïdes [GAU 1992]. Nous rappelons ici quelques notions à ce sujet. Ceci nous permettra par la suite d'adapter ce qui se fait pour les graphes d'événements temporisés à notre problématique. Pour plus de détails sur l'algèbre des dioïdes, on peut se référer à [GAU 1992], [BOI 2000] et [BAC 1992].

Les graphes d'événements sont composés de places et de transitions. Les places ont la propriété de n'avoir qu'une transition amont et qu'une transition aval. Les temporisations sont des durées minimales de tir ou de séjour. Nous nous intéressons ici aux graphes d'événement P-temporisés pour lesquels il y a dans chaque place un temps de séjour minimum. La figure 4.1 représente un tel graphe composé de six places et de cinq transitions. Seules les places P_1 et P_3 ont des temporisations non nulles respectivement égales à $\theta(P_1) = 3$ et $\theta(P_3) = 1$.

Deux types de variables peuvent être utilisés pour décrire l'état du système. Il s'agit, à l'emplacement de chaque transition, de variables compteurs, qui comptabilisent le passage des clients à un instant donné, et de variables dateurs, qui datent le passage d'un $n^{\text{ième}}$ client. Les compteurs sont fonctions du temps tandis que les dateurs sont fonctions d'un numéro de passage de client. On suppose que les jetons sont présents dans les places depuis l'instant $-\infty$, et qu'à l'instant initial zéro, aucune transition n'a été tirée. Dans ce cas, les dateurs sont positifs et les compteurs sont nuls jusqu'à l'instant initial zéro. On suppose connu le nombre de jetons présents dans chaque place avant l'instant initial, (cela revient à dire que le marquage initial est connu).

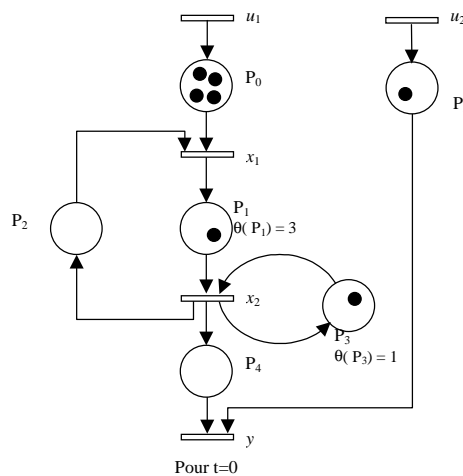


Figure 4.1 : Exemple de graphe d'événements P-temporisé

A partir de la représentation du graphe, du nombre initial de jetons, et des temporisations de chaque place, il est possible d'écrire un certain nombre d'équations traduisant les règles de fonctionnement de ce type de système. Sur la figure 4.1 toutes ces informations sont fournies puisque le marquage représenté est celui de l'instant initial. Nous pouvons donc écrire les inéquations (4.1) sur les dateurs, et les inéquations (4.2) sur les compteurs qui suivent. Les variables $x_1(n)$, $x_2(n)$, $u_1(n)$, $u_2(n)$ et $y(n)$ sont des dateurs. Par exemple $x_1(n)$ est la date du $n^{\text{ième}}$ passage de la transition x_1 . Les inéquations relatives aux dateurs sont alors :

$$\begin{aligned} x_1(n) &\geq \max(u_1(n-4), x_2(n)) \\ x_2(n) &\geq \max(x_1(n-1) + 3, x_2(n-1) + 1) \\ y(n) &\geq \max(u_2(n-1), x_2(n)) \end{aligned} \quad (4.1)$$

Les variables $x_1(t)$, $x_2(t)$, $u_1(t)$, $u_2(t)$ et $y(t)$ sont des compteurs. Par exemple $x_1(t)$ est le nombre de passages de la transition x_1 à l'instant t . Les inéquations relatives aux compteurs sont alors les suivantes :

$$\begin{aligned} x_1(t) &\leq \min(u_1(t) + 4, x_2(t)) \\ x_2(t) &\leq \min(x_1(t-3) + 1, x_2(t-1) + 1) \\ y(t) &\leq \min(u_2(t) + 1, x_2(t)) \end{aligned} \quad (4.2)$$

L'ensemble $\mathbb{R} \cup \{-\infty\}$, muni de l'opération max, notée \oplus , et de l'addition usuelle, notée \otimes , vérifie certaines propriétés et forme alors la structure algébrique d'un dioïde, notée \mathbb{R}_{\max} . L'élément neutre pour \oplus est $\varepsilon = -\infty$ et on a $e = 0$ pour élément neutre de \otimes .

Dans un dioïde, la relation d'ordre naturelle \succeq est définie par :

$$b \succeq a \Leftrightarrow a \preceq b \Leftrightarrow a \oplus b = b \quad (4.3)$$

Dans \mathbb{R}_{\max} , $a \succeq b$ équivaut à $a \oplus b = \max(a, b) = a$, c'est-à-dire $a \geq b$. Ici la relation \succeq coïncide avec l'ordre usuel. On peut ainsi écrire (4.1) dans \mathbb{R}_{\max} comme suit :

$$\begin{aligned} x_1(n) &\succeq u_1(n-4) \oplus x_2(n) \\ x_2(n) &\succeq 3x_1(n-1) \oplus 1x_2(n-1) \\ y(n) &\succeq u_2(n-1) \oplus x_2(n) \end{aligned} \quad (4.4)$$

On simplifie l'expression $3 \otimes x_1(n-1)$ en la remplaçant par $3x_1(n-1)$. Le système (4.4) peut aussi s'écrire sous la forme d'une écriture matricielle :

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ y(n) \end{bmatrix} \succeq \begin{bmatrix} \varepsilon & e \\ \varepsilon & \varepsilon \\ \varepsilon & e \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ 3 & 1 \\ \varepsilon & \varepsilon \end{bmatrix} \begin{bmatrix} x_1(n-1) \\ x_2(n-1) \end{bmatrix} \oplus \begin{bmatrix} e & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & e \end{bmatrix} \begin{bmatrix} u_1(n-4) \\ u_2(n-1) \end{bmatrix} \quad (4.5)$$

De même, l'ensemble $\mathbb{R} \cup \{+\infty\}$ muni de l'opération min, notée \oplus , et de l'addition usuelle, notée \otimes , vérifie certaines propriétés et forme la structure algébrique d'un dioïde, notée \mathbb{R}_{\min} . L'élément neutre pour \oplus est $\varepsilon = +\infty$ et dans \mathbb{R}_{\min} on a $e = 0$ pour élément neutre de \otimes . La relation d'ordre naturelle \succeq ne coïncide pas avec l'ordre usuel, car avoir

$a \succeq b$ équivaut à $a \oplus b = \min(a, b) = a$, c'est-à-dire $a \leq b$. On peut alors écrire (4.2) de la façon suivante dans \mathbb{R}_{\min} :

$$\begin{aligned} x_1(t) &\succeq 4u_1(t) \oplus x_2(t) \\ x_2(t) &\succeq 1x_1(t-3) \oplus 1x_2(t-1) \\ y(t) &\succeq 1u_2(t) \oplus x_2(t) \end{aligned} \quad (4.6)$$

Pour le graphe d'événement de la figure 4.1, il est possible de trouver les fonctions dateur et compteur avec tirage au plus tôt. Pour un système avec dateurs, tel que (4.4), ou avec compteurs, tel que (4.6), il suffit alors de trouver la solution minimale de ces systèmes d'inéquations linéaires dans leur dioïde respectif \mathbb{R}_{\max} et \mathbb{R}_{\min} , en utilisant la relation d'ordre \succeq . Cette solution peut être obtenue de façon explicite [GAU 1992].

Dans le cas particulier (non contraignant) d'un graphe d'événement où les temporisations sont toutes nulles, on peut écrire que les fonctions compteur en cas de tirage au plus tôt sont solutions minimales dans le dioïde \mathbb{R}_{\min} d'un système d'inéquations linéaires de la forme :

$$x(t) \succeq a x(t) \oplus b u(t) \quad (4.7)$$

où $x(t)$ et $u(t)$ sont des vecteurs de dimension respective n et m . Leurs composantes sont des compteurs, $x(t)$ correspond aux transitions précédées d'une place et $u(t)$ aux transitions précédées d'aucune place. Les matrices a et b appartiennent respectivement à $\mathbb{R}_{\min}^{n \times n}$ et $\mathbb{R}_{\min}^{n \times m}$. La solution minimale de (4.7) est donnée de façon explicite en utilisant l'opération étoile de Kleene :

$$x(t) = a^* b u(t) \quad \text{avec} \quad a^* = \bigoplus_{k \in \mathbb{N}} a^k \quad (4.8)$$

Cette solution revient à calculer pour chaque transition x_i le chemin vers une transition u_j sans prédécesseur où la somme de $u_j(t)$ et du marquage initial de chaque place rencontrée serait minimale. Puisque $a \in \mathbb{R}_{\min}^{n \times n}$ et qu'il n'existe pas de circuit négatif on a alors :

$$a^* = \text{Id} \oplus a \oplus a^2 \oplus \dots \oplus a^{n-1} \quad (4.9)$$

où Id est la matrice identité dont chaque composante vaut $e = 0$ sur la diagonale et $\varepsilon = +\infty$ ailleurs.

Lorsqu'il existe des circuits sans jetons, le graphe d'événements qui est sans temporisations n'est plus vivant, cependant dans ce cas particulier il y a une solution (4.8) qui ne met pas en évidence le fait qu'il y a un blocage. Un repérage des circuits sans jeton est donc nécessaire avant toute recherche de solution.

On souhaite exprimer F_i en fonction du vecteur d'état global X . Avec le formalisme de l'algèbre $(\min, +)$ et cette modélisation des graphes d'événements P-temporisés, on ne voit pas apparaître de façon immédiate les vecteurs d'état global X . Par ailleurs, les systèmes qui sont modélisés avec l'algèbre des dioïdes sont des graphes d'événements P-temporisés ; or ceux-ci ne permettent pas de modéliser directement les systèmes de production considérés, puisque l'on souhaite considérer des processus de fabrication comportant, par exemple, des serveurs avec des temps de service non déterministes.

Les principaux éléments que nous allons conserver sont les compteurs et quelques notations et résultats obtenus dans \mathbb{R}_{\min} . A partir de la valeur des compteurs il est possible d'en déduire la valeur du vecteur d'état global en fonction du temps, et l'algèbre $(\min,+)$ nous permettra d'exprimer les mécanismes de chacune des stations de synchronisation. La solution trouvée en (4.8) sera utile pour généraliser les résultats obtenus avec les politiques étudiées dans la section qui suit.

4.2 Calcul de F_i en fonction de X pour quelques politiques à flux tiré.

Le calcul de la valeur de F_i est effectué ici pour les politiques basestock, kanban, kanban étendu et kanban généralisé. Pour cela il est nécessaire d'utiliser des compteurs et quelques éléments de l'algèbre $(\min,+)$.

4.2.1 Notations et éléments nécessaires au calcul de F_i en fonction de X .

Les politiques des systèmes de production que nous étudions sont modélisées par des réseaux de files d'attente. On peut considérer qu'une file d'attente à capacité infinie est équivalente à une place avec temporisation nulle dans les graphes d'événements temporisés. De même les stations de synchronisation, si elles ne sont pas suivies d'un élément bloquant, ont un comportement similaire aux transitions des graphes d'événements. Dans un premier temps nous ne considérons que des lignes dont les processus de fabrication sont composés d'un stock tampon suivi d'une station monoserveur. Il n'y aura donc pas de blocage possible des stations de synchronisation qui précèdent les processus de fabrication. Pour connaître le nombre de pièces contenues dans les processus de fabrication on va utiliser des compteurs.

Sur chaque station de synchronisation il est possible de mettre un compteur. Les fonctions $u_i(t)$, $v_i(t)$, $v_N(t)$ et $d(t)$ sont les valeurs respectives des compteurs positionnés après un processus de fabrication F_i , avant ce même processus de fabrication, sur la station de synchronisation qui suit le stock de produits finis et sur celle qui génère l'arrivée de la demande (figure 4.2). D'autres compteurs peuvent être utilisés, ils correspondent à des stations de synchronisation utilisées par certaines politiques pour le contrôle des flux de production. Ceci existe par exemple pour les systèmes avec politique kanban généralisé (voir section 4.2.5).

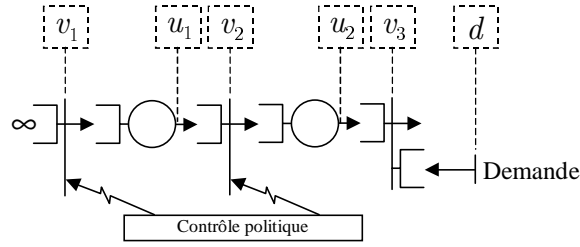


Figure 4.2 : Principaux compteurs utilisés pour une ligne de deux étages

Pour calculer F_i il suffit de déterminer sa valeur en fonction des compteurs $u_i(t)$ et $v_i(t)$ comme cela est fait dans la formule qui suit où les notations de l'algèbre (min,+) sont utilisées (l'addition usuelle est notée \otimes et la soustraction usuelle est notée $/$) :

$$F_i(t) = F_i(0) \otimes v_i(t) / u_i(t) \quad (4.10)$$

Pour exprimer F_i en fonction de X , il faudrait remplacer les compteurs de l'équation (4.10) par une expression des composantes de X . On sait exprimer, par la formule qui suit, la valeur de chaque composante de X en fonction des compteurs $u_i(t)$ et $d(t)$:

$$\begin{aligned} X_i(t) / X_i(0) &= u_i(t) / u_{i+1}(t) \text{ pour } 1 \leq i < N \\ X_N(t) / X_N(0) &= u_N(t) / d(t) \end{aligned} \quad (4.11)$$

En additionnant ces équations, on obtient d'autres formules qui seront utiles pour faire apparaître des sommes de composantes de X :

$$\begin{aligned} \bigotimes_{k=i}^j X_k(t) / \bigotimes_{k=i}^j X_k(0) &= u_i(t) / u_{j+1}(t) \text{ pour } 1 \leq i \leq j < N \\ \bigotimes_{k=i}^N X_k(t) / \bigotimes_{k=i}^N X_k(0) &= u_i(t) / d(t) \text{ pour } 1 \leq i \leq N \end{aligned} \quad (4.12)$$

L'équation (4.10) utilise le compteur $v_i(t)$ qui n'existe pas dans les formules (4.11) et (4.12). Il est donc nécessaire d'exprimer $v_i(t)$ par des compteurs $u_i(t)$ et $d(t)$ pour pouvoir obtenir F_i en fonction de X uniquement. Ce calcul va dépendre de la politique considérée.

La méthode que nous venons de décrire ci-dessus pour obtenir F_i en fonction de X est mise en œuvre dans les sections suivantes pour les politiques basestock, kanban, kanban étendu et kanban généralisé.

4.2.2 Calcul de F_i en fonction de X pour le basestock

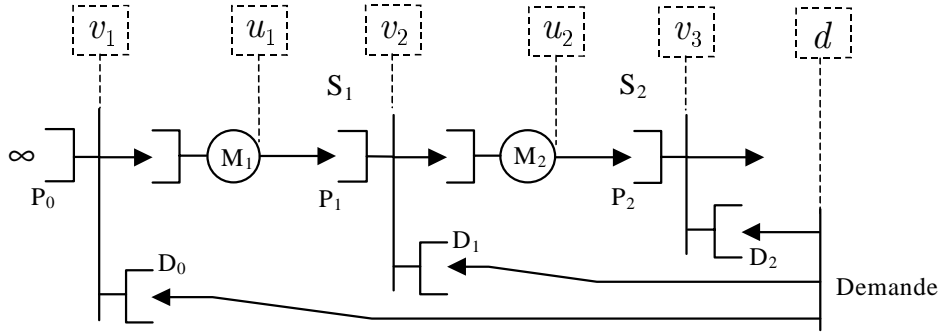


Figure 4.3 : Compteurs pour un basestock sur une ligne de deux mailles

Une représentation du basestock par un réseau de files d'attente est donnée sur la figure 4.3. La valeur du compteur $v_i(t)$ dépend du nombre de pièces arrivées en P_{i-1} et du nombre de demandes arrivées en D_{i-1} . Nous choisissons d'avoir à l'instant initial $P_{i-1} = S_{i-1}$ et $D_{i-1} = 0$;

on peut alors écrire ce qui suit avec les notations de l'algèbre $(\min,+)$:

$$\begin{aligned} v_i(t) &= S_{i-1} \otimes u_{i-1}(t) \oplus d(t) \quad \text{pour } i > 1 \\ v_1(t) &= d(t) \end{aligned} \quad (4.13)$$

Sachant que $F_i(0) = 0$ et en utilisant les formules (4.10) et (4.13) on trouve :

$$\begin{aligned} F_i(t) &= v_i(t) / u_i(t) = S_{i-1} \otimes u_{i-1}(t) / u_i(t) \oplus d(t) / u_i(t) \quad \text{pour } i > 1 \\ F_1(t) &= d(t) / u_1(t) \end{aligned} \quad (4.14)$$

Puis avec (4.12) et sachant que $X_i(0) = S_i$, on obtient :

$$\begin{aligned} F_i(t) &= X_{i-1}(t) \oplus \bigotimes_{k=i}^N S_k / \bigotimes_{k=i}^N X_k(t) \quad \text{pour } i > 1 \\ F_1(t) &= \bigotimes_{k=1}^N S_k / \bigotimes_{k=1}^N X_k(t) \end{aligned} \quad (4.15)$$

Dans ces formules la valeur de F_i est exprimée en fonction de X et des paramètres S_k du basestock. La représentation de la formule (4.15) avec des notations usuelles est la suivante :

$$\begin{aligned} F_i(t) &= \min(X_{i-1}(t), -X_i(t) - \dots - X_N(t) + S_i + \dots + S_N) \quad \text{pour } i > 1 \\ F_1(t) &= -X_1(t) - \dots - X_N(t) + S_1 + \dots + S_N \end{aligned}$$

4.2.3 Calcul de F_i en fonction de X pour le kanban

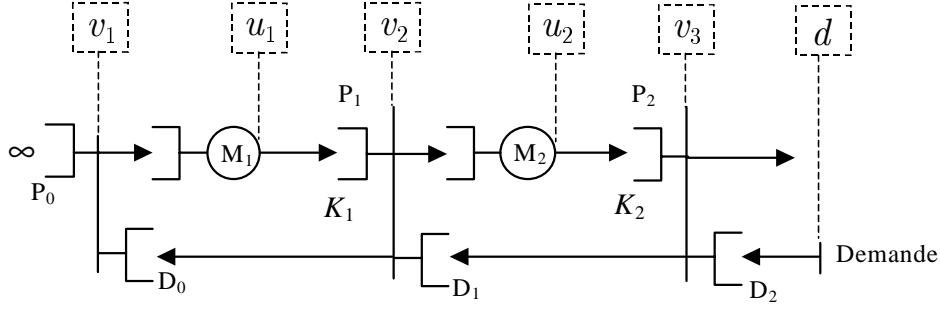


Figure 4.4 : Compteurs pour un kanban sur une ligne de deux mailles

On peut se référer à la figure 4.4 pour voir une représentation du kanban par un réseau de files d'attente. Avec cette politique la valeur du compteur $v_i(t)$ dépend du nombre de pièces arrivées en P_{i-1} et du nombre de kanbans arrivés en D_{i-1} , ou de demandes arrivées en D_N . Considérons un instant initial tel que $P_{i-1} = K_{i-1}$ et $D_{i-1} = 0$. Puisque les kanbans qui arrivent en D_{i-1} viennent de la station de synchronisation où se situe le compteur $v_{i+1}(t)$, on peut écrire ce qui suit avec les notations de l'algèbre $(\min, +)$:

$$\begin{aligned} v_1(t) &= v_2(t) \\ v_i(t) &= K_{i-1} \otimes u_{i-1}(t) \oplus v_{i+1}(t) \quad \text{pour } 1 < i < N + 1 \\ v_{N+1}(t) &= K_N \otimes u_N(t) \oplus d(t) \end{aligned} \quad (4.16)$$

Par des remplacements successifs on peut exprimer $v_i(t)$ avec les compteurs $d(t)$ et $u_j(t)$ comme ci-dessous.

$$v_i(t) = \bigoplus_{k=i}^{N+1} (K_{k-1} \otimes u_{k-1}(t)) \oplus d(t) \quad \text{pour } i > 1 \quad (4.17)$$

Sachant que $F_i(0) = 0$ et en utilisant les formules (4.10), (4.16) et (4.17), on trouve :

$$\begin{aligned} F_i(t) &= \bigoplus_{k=i}^{N+1} (K_{k-1} \otimes u_{k-1}(t) / u_i(t)) \oplus d(t) / u_i(t) \quad \text{pour } i > 1 \\ F_1(t) &= \bigoplus_{k=2}^{N+1} (K_{k-1} \otimes u_{k-1}(t) / u_1(t)) \oplus d(t) / u_1(t) \end{aligned} \quad (4.18)$$

Puis avec (4.12) et sachant que $X_{i-1}(0) = K_{i-1}$, on obtient :

$$\begin{aligned} F_i(t) &= X_{i-1}(t) \oplus K_i \oplus \bigoplus_{k=i+1}^N \left(\bigotimes_{l=i}^k K_l / \bigotimes_{l=i}^{k-1} X_l(t) \right) \oplus \bigotimes_{k=i}^N K_k / \bigotimes_{k=i}^N X_k(t) \\ F_1(t) &= K_1 \oplus \bigoplus_{k=2}^N \left(\bigotimes_{l=1}^k K_l / \bigotimes_{l=1}^{k-1} X_l(t) \right) \oplus \bigotimes_{k=1}^N K_k / \bigotimes_{k=1}^N X_k(t) \end{aligned} \quad (4.19)$$

Dans ces formules la valeur de F_i est exprimée en fonction de X et des paramètres K_k du kanban. La représentation de la formule (4.19) avec des notations usuelles est la suivante :

Pour $i > 1$,

$$F_i(t) = \min \left(\begin{array}{r} -X_N(t) - X_{N-1}(t) - \dots - X_i(t) \\ -X_{N-1}(t) - \dots - X_i(t) \\ \vdots \\ -X_i(t) \\ -X_i(t) \end{array} \begin{array}{l} + K_N + \dots + K_{i+1} + K_i, \\ + K_N + \dots + K_{i+1} + K_i, \\ \vdots + K_{i+1} + K_i, \\ K_{i+1} + K_i, \\ K_i, \end{array} \right. \\ \left. X_{i-1}(t) \right)$$

et

$$F_1(t) = \min \left(\begin{array}{r} -X_N(t) - X_{N-1}(t) - \dots - X_1(t) \\ -X_{N-1}(t) - \dots - X_1(t) \\ \vdots \\ -X_1(t) \\ -X_1(t) \end{array} \begin{array}{l} + K_N + \dots + K_2 + K_1, \\ + K_N + \dots + K_2 + K_1, \\ \vdots + K_2 + K_1, \\ K_2 + K_1, \\ K_1 \end{array} \right)$$

Il n'y a pas de simplification entre le premier et le second argument de la fonction min car X_N peut avoir une valeur négative.

4.2.4 Calcul de F_i en fonction de X pour le kanban étendu

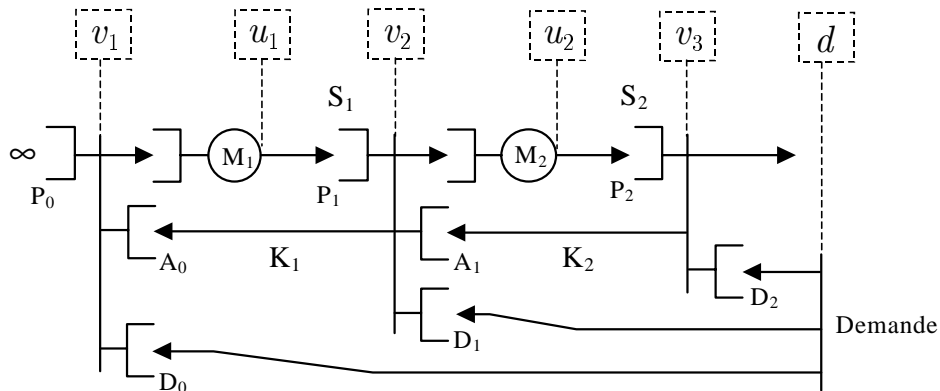


Figure 4.5 : Compteurs pour un kanban étendu sur une ligne de deux mailles

Le kanban étendu est un hybride du basestock et du kanban, il est représenté figure 4.5 pour une ligne de deux étages à l'aide d'un réseau de files d'attente. Avec cette politique, la valeur du compteur $v_i(t)$ dépend du nombre de pièces entrant en P_{i-1} , du nombre de kanbans arrivés en A_{i-1} et du nombre de demandes transmises en D_{i-1} . Les files d'attentes P_{i-1} , A_{i-1} et D_{i-1} sont respectivement alimentées par les stations de synchronisation où se trouvent les compteurs $u_{i-1}(t), v_{i+1}(t)$ et $d(t)$; on choisit d'avoir un instant initial tel que $P_{i-1} = S_{i-1}$, $A_{i-1} = K_i - S_i$ et $D_{i-1} = 0$. Ceci nous permet de déduire ce qui suit avec les notations de l'algèbre $(\min, +)$:

Par des remplacements itératifs, il est possible d'exprimer $w_i(t)$ avec les compteurs $d(t)$ et $u_j(t)$ comme ci-dessous.

$$w_i(t) = d(t) \oplus \bigoplus_{j=i}^N K_j \otimes u_j(t) \quad (4.26)$$

Cette valeur de $w_i(t)$ peut être introduite en (4.24), on trouve alors :

$$\begin{aligned} v_1(t) &= d(t) \oplus \bigoplus_{j=1}^N K_j \otimes u_j(t) \\ v_i(t) &= d(t) \oplus \bigoplus_{j=i}^N K_j \otimes u_j(t) \oplus S_{i-1} \otimes u_{i-1}(t) \quad \text{pour } N+1 > i > 1 \end{aligned} \quad (4.27)$$

Sachant que $F_i(0) = 0$ et en utilisant les formules (4.10) et (4.27) on trouve :

$$\begin{aligned} F_1(t) &= d(t)/u_1(t) \oplus \bigoplus_{j=1}^N (K_j \otimes u_j(t)/u_1(t)) \\ F_i(t) &= d(t)/u_i(t) \oplus \bigoplus_{j=i}^N (K_j \otimes u_j(t)/u_i(t)) \\ &\quad \oplus S_{i-1} \otimes u_{i-1}(t)/u_i(t) \quad \text{pour } N \geq i > 1 \end{aligned} \quad (4.28)$$

Avec les formules (4.12) et sachant que $X_i(0) = S_i$, on obtient :

$$\begin{aligned} F_1(t) &= \bigoplus_{j=2}^N \left(K_j \otimes \bigotimes_{k=1}^{j-1} (S_k / X_k(t)) \right) \oplus \bigotimes_{k=1}^N (S_k / X_k(t)) \oplus K_1 \\ F_i(t) &= \bigoplus_{j=i+1}^N \left(K_j \otimes \bigotimes_{k=i}^{j-1} (S_k / X_k(t)) \right) \oplus \bigotimes_{k=i}^N (S_k / X_k(t)) \oplus K_i \oplus X_{i-1}(t) \end{aligned} \quad (4.29)$$

pour $i > 1$

Dans ces formules la valeur de F_i est exprimée en fonction de X et des paramètres K_k et S_k du kanban généralisé. La représentation de la formule (4.29) avec des notations usuelles est la suivante :

Pour $i > 1$,

$$\begin{aligned} F_i(t) = \min \left(\begin{array}{ll} -X_N(t) - X_{N-1}(t) - X_{N-2}(t) - \dots - X_i(t) & + S_N + S_{N-1} + S_{N-2} + \dots + S_{i+1} + S_i, \\ -X_{N-1}(t) - X_{N-2}(t) - \dots - X_i(t) & + K_N + S_{N-1} + S_{N-2} + \dots + S_{i+1} + S_i, \\ -X_{N-2}(t) - \dots - X_i(t) & + K_{N-1} + S_{N-2} + \dots + S_{i+1} + S_i, \\ \vdots & \vdots + \dots + S_{i+1} + S_i, \\ -X_i(t) & + K_{i+1} + S_i, \\ & K_i, \\ & X_{i-1}(t) \end{array} \right) \end{aligned}$$

et

$$\begin{aligned} F_1(t) = \min \left(\begin{array}{ll} -X_N(t) - X_{N-1}(t) - X_{N-2}(t) - \dots - X_1(t) & + S_N + S_{N-1} + S_{N-2} + \dots + S_2 + S_1, \\ -X_{N-1}(t) - X_{N-2}(t) - \dots - X_1(t) & + K_N + S_{N-1} + S_{N-2} + \dots + S_2 + S_1, \\ -X_{N-2}(t) - \dots - X_1(t) & + K_{N-1} + S_{N-2} + \dots + S_2 + S_1, \\ \vdots & \vdots + \dots + S_2 + S_1, \\ -X_1(t) & + K_2 + S_1, \\ & K_1 \end{array} \right) \end{aligned}$$

4.2.6 Résultats résumés pour les quatre politiques

Pour chaque politique basestock, kanban, kanban étendu et kanban généralisé, le calcul de F_i en fonction de X a été possible et le résultat obtenu a toujours la même structure. Les formules trouvées ont à chaque fois la forme suivante :

$$\begin{aligned} F_1(t) &= \bigoplus_{j=0}^N \left(C_{(1,j)} / \bigotimes_{k=1}^j X_k(t) \right) \\ F_i(t) &= \bigoplus_{j=i-1}^N \left(C_{(i,j)} / \bigotimes_{k=i}^j X_k(t) \right) \oplus X_{i-1} \quad \text{pour } i > 1 \end{aligned} \quad (4.30)$$

Cette formulation sera dite canonique. Les paramètres $C_{(i,j)}$ sont des constantes à valeur réelle, ou ε l'élément neutre de \oplus dans \mathbb{R}_{\min} . Ils varient suivant la politique considérée. On a pour le basestock :

$$C_{(i,N)} = \bigotimes_{k=i}^N S_k \text{ et } C_{(i,j)} = \varepsilon = +\infty \text{ avec } i-1 \leq j < N$$

pour le kanban :

$$C_{(i,N)} = \bigotimes_{k=i}^N K_k \text{ et } C_{(i,j)} = \bigotimes_{k=i}^{j+1} K_k \text{ avec } i-1 \leq j < N$$

pour le kanban étendu :

$$C_{(i,N)} = \bigotimes_{k=i}^N S_k \text{ et } C_{(i,j)} = \bigotimes_{k=i}^{j+1} K_k \text{ avec } i-1 \leq j < N$$

et pour le kanban généralisé :

$$C_{(i,N)} = \bigotimes_{k=i}^N S_k \text{ et } C_{(i,j)} = K_{j+1} \otimes \bigotimes_{k=i}^j S_k \text{ avec } i-1 \leq j < N$$

Notons que pour le kanban étendu et le kanban généralisé on retrouve le kanban quand $K_i = S_i$ et le basestock quand $K_i = +\infty$.

Quel que soit le paramétrage $C_{(i,j)}$, la valeur de $F_i(t)$, pour tout t , ne doit jamais être négative. Ainsi les états globaux X sont admissibles si, et seulement si, la valeur de $F_i(X)$ est positive ou nulle. Cela correspond à un polyèdre à valeurs entières délimité par les inéquations suivantes :

$$\sum_{k=i}^j X_k \leq C_{(i,j)} \text{ pour tout } i \text{ et } j \text{ tels que } 1 \leq i \leq j \leq N \text{ avec } X \in \mathbb{N}^{N-1} \times \mathbb{Z} \quad (4.31)$$

Il est possible d'obtenir une formulation canonique des politiques pour beaucoup d'autres systèmes. Nous montrons en particulier qu'un ensemble de politiques de gestion de production à flux tiré, modélisées par réseaux de files d'attente, vérifient cette formulation.

- Une file d'attente P_i pour stocker les produits intermédiaires. Le stock de produits bruts est P_0 et cette file d'attente n'est jamais vide. On pose $P_0 = \infty$.
- Une station de synchronisation v_{i+1} (associée au compteur de valeur $v_{i+1}(t)$) dont une des entrées provient de la file d'attente P_i et dont une des sorties est dirigée sur le stock tampon de F_{i+1} , ou vers la sortie si $i = N$.
- La partie contrôle du système est constituée par :
 - Des stations de synchronisation $w_{(i,j)}$ (associées aux compteurs de valeur $w_{(i,j)}(t)$) dont les sorties sont dirigées vers des files d'attente $A_{(l,k)}$. On a $1 \leq l \leq i \leq N$.
 - Des files d'attente $A_{(i,j)}$ attachées aux stations de synchronisations v_i ou $w_{(i,k)}$. Ces files d'attente sont alimentées par les stations de synchronisation $u_k, v_{k+1}, w_{(k,l)}$ ou d avec $1 \leq i \leq k \leq N$.
 - Une file d'attente A_{N+1} se situe en entrée de v_{N+1} et est alimentée par d, la station de synchronisation d'où provient la demande de produits finis.

□

Proposition 4.1 :

Pour un système de production modélisé par un réseau de files d'attente du même type que celui de la figure 4.7 et respectant la définition 4.1, il existe une formulation canonique de F_i en fonction de X .

□

Preuve :

Pour trouver F_i en fonction de X , il suffit d'utiliser (4.10) et (4.12). La seule difficulté consiste à exprimer $v_i(t)$ en fonction des compteurs $u_j(t)$ et $d(t)$.

Pour $1 \leq i \leq N$ les seules files d'attente présentes en entrée de v_i sont P_{i-1} et éventuellement $A_{(i,j)}$ pour certains indices j qui forment un ensemble J_i . Pour $i = 1$ on a $P_0 = \varepsilon = +\infty$, pour $i > 1$ on a un stock P_{i-1} qui est précédé de u_{i-1} . Quant aux files d'attente $A_{(i,j)}$, elles sont alimentées soit par $u_k, w_{(k,l)}, v_k, v_{N+1}$, ou d, avec k compris entre i et N . Toutes ces stations de synchronisation ont un compteur associé, noté $z_{(i,j)}(t)$, avec i et j les indices de $A_{(i,j)}$. On a donc

$$\begin{aligned}
v_i(t) &= P_{i-1}(0)u_{i-1}(t) \oplus \bigoplus_{j \in J_i} \left(A_{(i,j)}(0)z_{(i,j)}(t) \right) \quad \text{pour } 1 < i \leq N \\
v_1(t) &= \bigoplus_{j \in J_1} \left(A_{(1,j)}(0)z_{(1,j)}(t) \right)
\end{aligned} \tag{4.32}$$

Considérons le réseau de files d'attente correspondant à la partie contrôle et constitué des files $P_j, A_{(j,k)}, A_{N+1}$, des stations de synchronisations $u_j, w_{(j,l)}, v_j, v_{N+1}$ et d, pour j allant de i à N (voir figure 4.8). En supprimant les processus de fabrication, les stations de synchronisation u_j se retrouvent, comme c'est le cas pour d, sans prédécesseur. De plus

ce sous-réseau de files d'attente, qui vient d'être défini, est indépendant du reste du système car pour $j < i$, il n'existe aucun arc provenant des stations de synchronisation u_j , $w_{(j,l)}$ ou v_j , et allant vers les files P_i , $A_{(i,k)}$ ou A_{N+1} . Aucune file d'attente extérieure à ce sous-réseau n'est attachée à une station de synchronisation de ce sous-réseau.

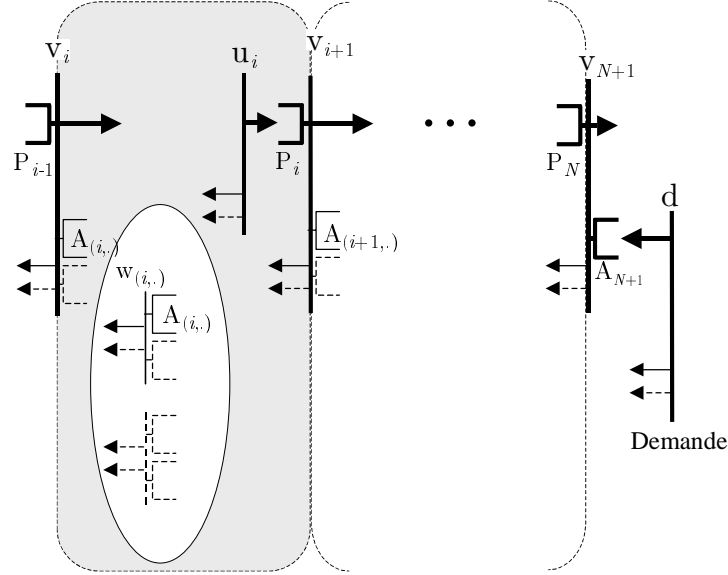


Figure 4.8 : Sous-réseau de files d'attente extrait la figure 4.7

On peut transformer ce sous-réseau en un graphe d'événements équivalent et il suffit pour cela, de remplacer chaque file d'attente par une place avec temporisation nulle et chaque station de synchronisation par une transition. Par commodité, lors de cette transformation les noms des différents éléments restent inchangés.

Ce graphe d'événements a des temporisations nulles, on peut donc écrire le système d'inéquations identique à (4.7) :

$$x(t) \succeq a x(t) \oplus b u(t)$$

Pour j allant de i à N , le vecteur $x(t)$ a pour composante les compteurs $w_{(j,l)}$, v_j et v_{N+1} ; et le vecteur $u(t)$ a pour composante les compteurs u_j et d . Les matrices a et b se construisent ligne par ligne avec une simple mise en équation de chaque transition. La solution minimale de ce système d'inéquations permet d'obtenir la valeur des compteurs en cas de franchissement des transitions au plus tôt. La solution minimale (cf. (4.8)) est :

$$x(t) = a^* b u(t) \quad \text{avec} \quad a^* = \bigoplus_{k \in \mathbb{N}} a^k$$

Les compteurs $z_{(i,j)}(t)$ sont donc définis en fonction de u_j , avec j allant de i à N , et de d .

On peut écrire

$$z_{(i,j)}(t) = \bigoplus_{k=i}^N (c_{(i,j,k)} u_k(t)) \oplus c_{(i,j,0)} d(t),$$

où $c_{(i,j,k)}$ est une constante positive éventuellement égale à l'élément neutre $\varepsilon = +\infty$.

En utilisant (4.10), et sachant que $F_i(0) = 0$ on trouve

$$F_i(t) = P_{i-1}(0)u_{i-1}(t)/u_i(t) \oplus \left(\bigoplus_{j \in J_i} \left(A_{(i,j)}(0) \left(\bigoplus_{k=i}^N (c_{(i,j,k)} u_k(t)/u_i(t)) \oplus c_{(i,j,0)} d(t)/u_i(t) \right) \right) \right)$$

et pour $i = 1$ on a

$$F_1(t) = \bigoplus_{j \in J_1} \left(A_{(1,j)}(0) \left(\bigoplus_{k=1}^N (c_{(1,j,k)} u_k(t)/u_1(t)) \oplus c_{(1,j,0)} d(t)/u_1(t) \right) \right)$$

Avec (4.12), on peut remplacer dans les formules ci-dessus $u_i(t)/u_{j+1}(t)$ et $u_i(t)/d(t)$. On trouve ainsi la formulation dite canonique.

□

Pour déterminer cette formulation nous donnons ci-dessous un algorithme [BOL 2000]. Il utilise le fait que la solution explicite (4.8) peut être calculée par une recherche de plus court chemin [GAU 1992].

Algorithme 1 :

// Détermination de $F_i(X)$ pour un système défini en accord avec la proposition 4.1.

// Construction d'un graphe G temporaire utilisé pour les calculs

- 1 $G =$ Graphe nul
- 2 **Ajouter** à G un **sommet** pour chaque transition v_i , u_i et d , et nommer chacun comme la transition qui lui est associée.
- 3 **Ajouter** à G un **arc** d'un sommet à un autre chaque fois qu'un arc existe entre les stations de synchronisation associées. Ne pas ajouter d'arc allant de v_i à u_i . Le poids de chaque arc dans G est donné par le nombre initial de jetons présents dans la file d'attente située à l'extrémité de l'arc associé.
- 4 **Supprimer** tout cycle de longueur nulle dans le graphe G .

// Calcul de F_i

5 **Pour** i allant de 1 à N faire :

6 **Pour** j allant de i à N faire :

7 $C_{(i,j)} = \left(\bigotimes_{k=i}^{j-1} P_k(0) \right) \otimes$ (le chemin de poids minimal allant de u_j à v_i dans G).

8 **Fin pour**

9 $C_{(i,N+1)} = \left(\bigotimes_{k=i}^N P_k(0) \right) \otimes$ (le chemin de poids minimal allant de d à v_i dans G).

10 **Si** $i > 1$ **alors**

11 Imprimer $F_i = \bigoplus_{j=i-1}^N \left(C_{(i,j)} / \bigotimes_{k=i}^j X_k \right) \oplus X_{i-1}$

12 **Sinon**

13 Imprimer $F_1 = \bigoplus_{j=0}^N \left(C_{(1,j)} / \bigotimes_{k=1}^j X_k \right)$

14 Fin si

15 Fin pour

On peut utiliser l'algorithme de Dijkstra pour calculer les chemins de poids minimal car les pondérations sont toutes positives. Il faut aussi vérifier qu'en ajoutant un arc de poids $F_i(0)$ allant de v_i à u_i , il n'y a pas de cycle de poids nul. Dans un tel cas le système n'est pas vivant. Ceci se produit par exemple avec un système kanban dont l'un des paramètres K_i est nul.

Sur la figure 4.9, l'algorithme 1 est illustré pour une politique CONWIP/kanban [BON 1996]. De gauche à droite on peut voir le réseau de files d'attente de la politique, le graphe G et le calcul de F_1 et F_2 .

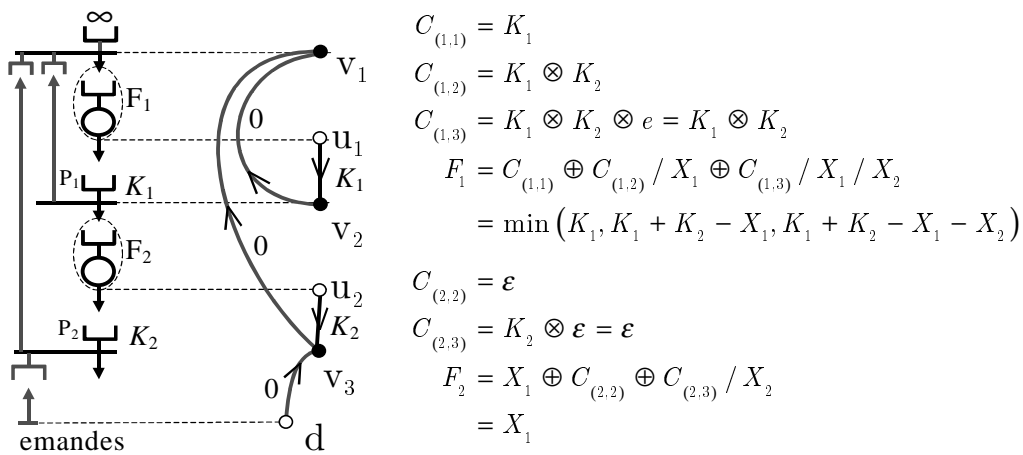


Figure 4.9 : Algorithme 1 pour une politique CONWIP/kanban

Nous avons calculé la valeur de F_i en fonction de X pour un processus de fabrication constitué d'un stock tampon suivi d'une station. Dans les sections qui suivent, nous allons adapter ces calculs à d'autres types de processus de fabrication, lorsque cela est possible, et dans chaque cas interpréter le rôle joué par la valeur F_i .

4.4 Utilisation de F_i fonction de X pour différents processus de fabrication

Dans cette section on montre comment utiliser $F_i(X)$ pour différents types de processus de fabrication. L'interprétation du rôle joué par la valeur $F_i(X)$ n'est pas toujours la même. De plus, suivant les processus de fabrication, les précédentes techniques de calcul ne s'appliquent pas toujours directement car il peut être nécessaire de faire quelques modifications.

4.4.1 Une station monoserveur avec stock tampon

Pour une station monoserveur la valeur de $F_i(X)$ permet de savoir en utilisant (3.4) (c'est-à-dire, $M_i = \min(1, F_i)$) si la station M_i sert une pièce ou n'en sert pas. Dans ce cas, on peut donc en fonction de X , commander la marche et l'arrêt de la station. On parle alors de politique de pilotage plutôt que de politique de contrôle du système de production.

Ce cas a été étudié en [VEA 1992] par Veatch et Wein. Ils ont montré que sous certaines conditions, le pilotage optimal des stations s'exprime par de simples zones de commande délimitées par des courbes appelées "switching functions". Ces courbes sont décrites en [VEA 1994] pour plusieurs politiques usuelles dont le kanban et le basestock sur des lignes de deux étages. La figure 4.10 montre de telles courbes pour les politiques basestock, kanban, kanban étendu et kanban généralisé pour une ligne de deux étages. La courbe de commande de la machine 2 est en gris et la courbe de commande de la machine 1 est en noir. Dans les zones hachurées, les deux stations de la ligne marchent, en dehors au moins une des stations est à l'arrêt suivant la ou les courbes de commande franchies.

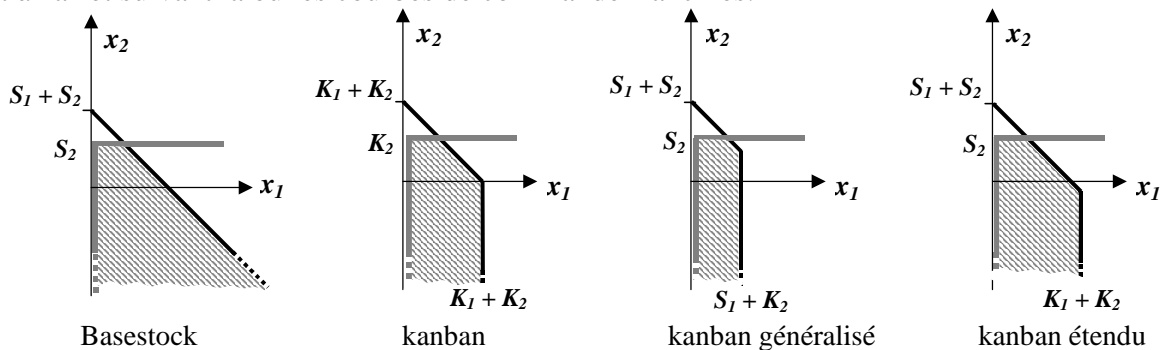


Figure 4.10 : Courbe de commande de plusieurs politiques pour une ligne de deux mailles

Pour une politique exprimée avec la formulation dite canonique, ces courbes de commande correspondent alors aux facettes du polyèdre (4.31) qui décrit l'ensemble des états admissibles. Chaque facette correspond à l'arrêt d'au moins un des processus de fabrication et à l'intérieur du domaine toutes les stations sont en marche. Les informations fournies par F_i fonction de X sont suffisantes pour connaître l'état de marche, ou d'arrêt, des machines, pour d'autres types de processus de fabrication cela n'est pas toujours vrai.

4.4.2 Une station monoserveur sans stock tampon

Dans ce cas, comme précédemment il est possible de déterminer la marche et l'arrêt des stations en fonction de X car F_i vaut soit un, soit zéro, et la station i est alors respectivement soit en marche, soit à l'arrêt. Cependant les calculs effectués précédemment pour F_i ne sont plus valables, car l'absence de stock tampon peut créer des blocages, par saturation des monoserveurs. On a au plus une pièce en F_i , donc le nombre de franchissements en $v_i(t)$ est

contraint, à ne pas dépasser $u_i(t)+1$, si l'on suppose que F_i est nul à l'instant zéro. On peut donc calculer F_i par une légère modification de l'algorithme 1.

A la suite de la ligne 3, on insère :

« Ajouter à G un arc de poids 1 allant de u_i à v_i »

On remplace la ligne 11 par

$$\ll \text{Imprimer } F_i = 1 \oplus \bigoplus_{j=i-1}^N \left(C_{(i,j)} / \bigotimes_{k=i}^j X_k \right) \oplus X_{i-1} \gg$$

et la ligne 13 par

$$\ll \text{Imprimer } F_1 = 1 \oplus \bigoplus_{j=0}^N \left(C_{(1,j)} / \bigotimes_{k=1}^j X_k \right) \gg$$

Par exemple pour le basestock on trouve :

$$F_i(t) = 1 \oplus X_{i-1}(t) \oplus \bigotimes_{k=i}^N S_k / \bigotimes_{k=i}^N X_k(t) \quad \text{pour } N \geq i > 1$$

$$F_1(t) = 1 \oplus \bigotimes_{k=1}^N S_k / \bigotimes_{k=1}^N X_k(t)$$

Dans le cas du basestock, le fait d'utiliser des processus de fabrication composés uniquement d'un monoserveur ne modifie presque pas la politique. Lorsqu'il y a des stocks tampon dans les processus de fabrication, le nombre de pièces qu'ils contiennent n'est pas borné avec la politique basestock, dans notre cas ces stocks tampon n'existent pas, c'est alors le stock de produits intermédiaires qui joue ce rôle et celui-ci n'est plus borné. Effectivement, dans ces formules, F_i est bornée mais on observe aussi que X_i ne l'est pas, donc P_{i-1} ne l'est pas non plus. Le problème des encours infinis a simplement été déplacé.

Pour le kanban on trouve :

$$F_i(t) = X_{i-1}(t) \oplus 1 \oplus \bigoplus_{k=i}^{N-1} \left(1 \otimes \bigotimes_{l=i}^k K_l / \bigotimes_{l=i}^k X_l(t) \right) \oplus \bigotimes_{k=i}^N K_k / \bigotimes_{k=i}^N X_k(t) \quad \text{pour } i > 1$$

$$F_1(t) = 1 \oplus \bigoplus_{k=1}^{N-1} \left(1 \otimes \bigotimes_{l=1}^k K_l / \bigotimes_{l=1}^k X_l(t) \right) \oplus \bigotimes_{k=1}^N K_k / \bigotimes_{k=1}^N X_k(t)$$

Ici la suppression du stock d'en-cours modifie la dynamique du système kanban car la remontée des informations est interrompue par ces blocages. Cette modification de la politique kanban due à l'absence de files d'attente tampon est connue sous le nom de système kanban modifié [DIM 1990] (voir la section 2.5.3).

4.4.3 Une station multiserveur avec tampon

Pour ce type de station il est possible de connaître le nombre de serveurs actifs à partir de F_i . Si la station est un monoserveur alors connaître le nombre de serveurs actifs signifie que l'on sait si la station est en marche ou à l'arrêt, ceci a déjà été traité précédemment.

Appelons n_i le nombre de serveurs pour la station. M_i correspond au nombre de pièces présentes dans cette station. Chaque pièce présente dans la station est associée à un serveur qui devient actif, donc M_i correspond au nombre de serveurs actifs dans la station et sa valeur est comprise entre 0 et n_i . En prenant les formules de F_i obtenues pour F_i possédant un encours, on trouve facilement ce nombre de serveurs actifs :

$$M_i = \min(n_i, F_i)$$

Pour le cas d'un serveur infini, la valeur de n_i est infinie, donc

$$M_i = F_i$$

Le domaine où l'état global est admissible se situe aux points où les fonctions F_i de X sont positives ou nulles. Il s'agit du polyèdre défini en (4.31). Sur le bord de ce polyèdre M_i est nul et à l'intérieur M_i augmente progressivement au fur et à mesure que l'on s'éloigne des bords. Cette augmentation de la valeur de M_i est limitée par une valeur seuil qui est n_i ou une autre constante qui peut correspondre à un nombre de kanbans par exemple.

Pour le cas monoserveur ($n_i=1$) les bords du polyèdre défini en (4.31) délimitent une zone de marche et une zone d'arrêt des machines ; pour une station multiserveur ($n_i>1$), ce polyèdre délimite une zone où les stations sont en marche mais cependant à l'intérieur de cette zone le nombre de serveurs en marche pour une station multiserveur M_i est égal à M_i . Donc pour $n_i>1$, le nombre de serveurs en marche dans une station ne passe pas brutalement de zéro à n_i .

4.4.4 Plusieurs stations monoserveur en série avec stock tampon

Lorsque le processus de fabrication est composé de n_i stations en série avec stock tampon, les calculs de F_i effectués dans l'algorithme 1 restent valables. Cependant les résultats obtenus n'apporteront pas d'informations suffisante sur la marche et l'arrêt des stations pour pouvoir envisager leur pilotage en fonction du vecteur X .

Si F_i est nul on sait qu'aucune station ne fonctionne. Si F_i est supérieur ou égal à 1 alors au moins une station est en marche ; on sait aussi qu'on ne peut pas avoir plus de F_i ou n_i serveurs en marche. Cependant on ne peut pas savoir précisément combien de monoserveurs sont en marche (voir figure 4.11), on sait seulement qu'il y en a entre :

$$\min(1, F_i) \text{ et } \min(n_i, F_i)$$

Pour déterminer les stations actives de F_i il serait nécessaire de découper le processus de fabrication en ajoutant n_i-1 variables d'état comme cela a été fait dans l'exemple de la figure 3.13 page 58.

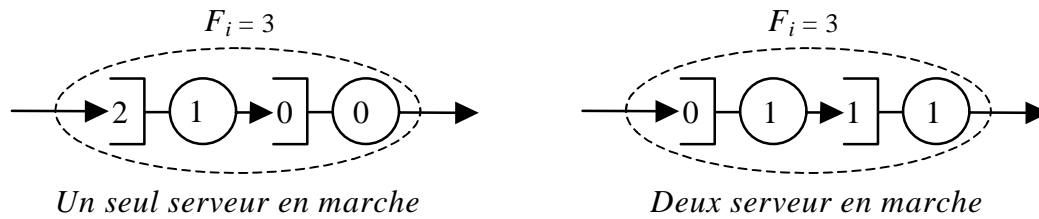


Figure 4.11 : Valeurs des F_i identiques et marche des serveurs différente pour deux processus de fabrication ayant deux stations monoserveur en série.

4.5 Application aux systèmes avec production par lots

On s'intéresse ici à des processus de fabrication composés d'une file d'attente tampon et d'une station servant par lots de taille Q_i . La prise en compte de la production par lots augmente le nombre des politiques de gestion de production envisageables. Dans la littérature, les politiques (Q, r) ou (s, S) de production par lots sont souvent citées. Pour des lots d'une pièce, ces politiques sont réputées être équivalentes au basestock [AXS 1993].

Parmi les politiques (Q, r) , on distingue celles de type échelon et celles de type installation (voir les sections 2.5.1 et 2.5.2). On notera (Q, r) par (Q, r^e) pour une politique de type échelon et (Q, r^i) pour une politique de type installation. Il a été montré dans [AXS 1993] que la politique (Q, r^i) de type installation était un cas particulier d'une politique (Q, r^e) de type échelon. Nous nous intéresserons donc principalement à cette dernière. Un exemple de kanban avec lots sera aussi traité.

4.5.1 Calcul de F_i pour une politique (Q, r^e) de type échelon

Une politique (Q, r^e) de type échelon peut être modélisée par un réseau de files d'attente équivalent comme le montre la figure 4.12. Sur cette figure les barres obliques sont des transitions qui ne sont franchissables que par des lots ayant la taille requise.

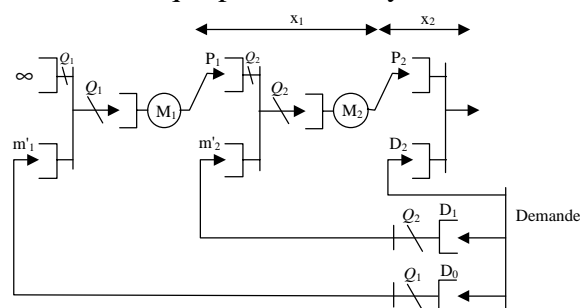


Figure 4.12 : Système équivalent à la politique (Q, r^e) de type échelon pour 2 étages

Une politique (Q, r^e) de type échelon consiste à envoyer un ordre de production pour l'étage n chaque fois que le niveau de l'échelon n (noté I_n et défini en (4.33)) est égal ou

inférieur à r_n^e . L'ordre de production consiste alors à demander m_n lots supplémentaires de taille Q_n pour que le niveau de l'échelon soit juste supérieur à r_n^e . Parmi les m_n lots supplémentaires demandés, un certain nombre peuvent être pris en compte instantanément. Cependant lorsqu'il n'y a pas suffisamment de produits en P_{n-1} , un nombre m'_n d'ordres de production de lots de taille Q_n peuvent rester en attente devant F_n . Le niveau I_n de l'échelon n est égal à la somme des pièces présentes dans les étages n à N , moins les demandes non servies et plus $m'_n Q_n$, le nombre de pièces autorisées à être produites mais qui ne sont pas encore allées en F_i :

$$I_n = F_n + \sum_{k=n}^N X_k + m'_n \cdot Q_n \quad (4.33)$$

A chaque arrivée de demande, I_n diminue de un ($Q_{N+1}=1$). Par contre I_n n'augmente pas tant qu'un nouvel ordre de production n'est pas lancé. La variable décroît donc lentement jusqu'à atteindre le niveau r_n^e et à cet instant un ordre de fabrication est lancé ($m_n=1$). On a alors $I_n = r_n^e + Q_n$. Etant donnée cette dynamique, on est certain que I_n est compris entre $r_n^e + 1$ et $r_n^e + Q_n$. On peut donc définir le nombre d'ordres de fabrication non pris en compte comme :

$$m'_n \cdot Q_n > r_n^e - F_n - \sum_{k=n}^N X_k \geq m'_n \cdot Q_n - Q_n \quad \text{avec } m'_n \text{ entier}$$

donc, avec $E[x]$ partie entière de x , et sachant que F_n/Q_n est entier on a

$$m'_n = E \left[\left(r_n^e - F_n - \sum_{k=n}^N X_k \right) / Q_n + 1 \right] = E \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] + 1 - \frac{F_n}{Q_n} \quad (4.34)$$

Si Q_n n'est pas multiple de Q_{n+1} alors le nombre de pièces en P_n n'est pas souvent multiple de Q_{n+1} , on ne peut pas alors transformer toutes les pièces de P_n en lots de taille Q_{n+1} pour satisfaire la demande sur les l'étages situés plus en aval. Dans [AXS 1993], il est indiqué qu'il faut choisir Q_n multiple de Q_{n+1} . En partant avec un système où les stocks initiaux sont vides, on obtient alors une valeur de P_{n-1} multiple de Q_n . Ainsi, avoir $m'_n > 0$ signifie que P_{n-1} est nul. On trouve alors :

$$m'_n > 0 \Rightarrow X_{n-1} = P_{n-1} + F_n = F_n$$

puis en utilisant (4.34) on obtient :

$$m'_n > 0 \Leftrightarrow r_n^e + Q_n - F_n - \sum_{k=n}^N X_k \geq Q_n \Rightarrow r_n^e - \sum_{k=n}^N X_k - X_{n-1} + Q_n \geq Q_n$$

Puisque $X_{n-1} \geq F_n$ on a :

$$m'_n = 0 \Leftrightarrow r_n^e + Q_n - F_n - \sum_{k=n}^N X_k < Q_n \Rightarrow r_n^e - \sum_{k=n}^N X_k - X_{n-1} + Q_n < Q_n$$

Ainsi avec (4.34) en sachant que X_{n-1} est multiple de Q_n on trouve :

$$m'_n = \max \left(0, \mathbb{E} \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] - \frac{X_{n-1}}{Q_n} + 1 \right) \quad (4.35)$$

Avec (4.34), on en déduit :

$$\begin{aligned} \frac{F_n}{Q_n} &= \mathbb{E} \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] + 1 - \max \left(0, \mathbb{E} \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] - \frac{X_{n-1}}{Q_n} + 1 \right) \\ &= \min \left(\mathbb{E} \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] + 1, \frac{X_{n-1}}{Q_n} \right) \end{aligned}$$

et :

$$F_n = \min \left(Q_n \cdot \mathbb{E} \left[\left(r_n^e - \sum_{k=n}^N X_k \right) / Q_n \right] + Q_n, X_{n-1} \right)$$

En utilisant la fonction $\mathbb{E}_{Q_n} [x] = Q_n \cdot \mathbb{E} \left[\frac{x}{Q_n} \right]$ on obtient :

$$F_n = \mathbb{E}_{Q_n} \left[\min \left(r_n^e + Q_n - \sum_{k=n}^N X_k, X_{n-1} \right) \right]$$

En définissant, $S_n = r_n^e + Q_n - r_{n+1}^e - Q_{n+1}$ pour $n < N$ et $S_N = r_N^e + Q_N$ ou de façon équivalente en posant :

$$r_n^e + Q_n = \sum_{k=n}^N S_k \text{ pour } n \leq N \quad (4.36)$$

on obtient la formule (4.37) très proche de (4.15) utilisée pour le basestock.

$$\begin{aligned} F_n &= \mathbb{E}_{Q_n} \left[\min \left(\sum_{k=n}^N S_k - \sum_{k=n}^N X_k, X_{n-1} \right) \right] \\ F_1 &= \mathbb{E}_{Q_1} \left[\sum_{k=1}^N S_k - \sum_{k=1}^N X_k \right] \end{aligned} \quad (4.37)$$

Donc si $\sum_{k=n}^N X_k > \sum_{k=n}^N S_k - Q_n$ ou $X_{n-1} < Q_n$ alors le processus de fabrication n est à

l'arrêt. Ainsi il est possible de définir des zones de marche et d'arrêt. Pour des lots de taille un on retrouve exactement la formule du basestock.

4.5.2 Calcul de F_i pour une politique (Q, r^i) de type installation

Une politique (Q, r^i) de type installation peut être modélisée par un réseau de files d'attente équivalent comme le montre la figure 4.13.

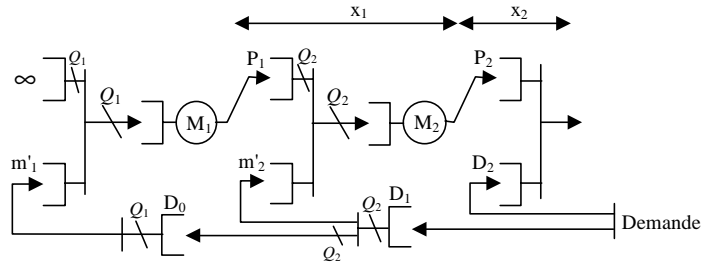


Figure 4.13 : Système équivalent à la politique (Q,r) de type installation à 2 étages

Une politique (Q, r^i) de type installation fonctionne comme la politique (Q, r^e) de type échelon sauf que le contrôle de l'inventaire porte sur i_n qui vérifie $I_n = \sum_{k=n}^N i_k$. Chaque fois que $i_n \leq r_n^i$, une autorisation de produire des lots de taille Q_n est transmise. Pour la politique (Q, r^i) une modélisation équivalente par réseaux de files d'attente est possible, la figure 4.13 en est un exemple pour une ligne de deux étages. Dans [AXS 1993], il est montré que la politique (Q, r^i) était identique à une politique (Q, r^e) pour laquelle certaines conditions sur l'état initial (à $t = 0$) étaient posées. Ainsi la formule (4.37) reste valable à condition que :

$$I_{n-1}(0) - r_{n-1}^e - I_n(0) + r_n^e + Q_n \quad (4.38)$$

soit positif et multiple de Q_n pour n allant de 2 à N .

On note « mod » le modulo. En utilisant (4.33), (4.34) et la propriété $(x - E_{Q_n}(x) = x \bmod Q_n)$ on a (4.38) égale à :

$$\left(r_n^e - \sum_{k=n}^N X_k(0) \right) \bmod Q_n - \left(r_{n-1}^e - \sum_{k=n-1}^N X_k(0) \right) \bmod Q_{n-1} + Q_{n-1}$$

qui avec (4.36) devient :

$$\left(\sum_{k=n}^N (S_k - X_k^0) \right) \bmod Q_n - \left(\sum_{k=n-1}^N (S_k - X_k^0) \right) \bmod Q_{n-1} + Q_{n-1}$$

Cela montre que (4.38) est bien positive. En utilisant la propriété $((A \bmod Q_n - B \bmod Q_{n-1}) \bmod Q_n = (A - B) \bmod Q_n)$ on trouve ainsi qu'il suffit que la formule :

$$X_{n-1}(0) - S_{n-1}$$

soit multiple de Q_n pour que (4.38) le soit aussi. La politique (Q, r^i) s'exprime donc par la formule (4.37) à la condition que $X_{n-1}(0) - S_{n-1}$ soit multiple de Q_n .

Puisqu'une variable X_{n-1} sera toujours augmentée par Q_{n-1} multiple de Q_n ou diminuée de Q_n et comme F_n est multiple de Q_n alors pour ne pas avoir de stock résiduel inutile en P_{n-1} il est nécessaire que X_{n-1} et S_{n-1} soient multiples de Q_n . On peut par exemple prendre $X_{n-1}(0) = S_{n-1}$ avec S_{n-1} multiple de Q_n .

Cette contrainte sur l'état initial permet de synchroniser la demande d'un lot de produits intermédiaires dans un étage n avec celles provenant de l'étage $n-1$. Ce phénomène de synchronisation des autorisations de production peut s'observer sur la figure 4.13.

Pour qu'il y ait synchronisation de ces autorisations de production il suffit que l'augmentation de la quantité $(Q_n m'_n + F_n)$ soit toujours provoquée par une augmentation de $(Q_{n+1} m'_{n+1} + F_{n+1})$. En utilisant (4.34) on observe que $(Q_n m'_n + F_n)$ ne peut être

augmenté que par une diminution de 1 de la valeur X_N quand $\sum_{k=n}^N (S_k - X_k)$ est multiple de

Q_n . Or $S_n - X_n$ est toujours multiple de Q_{n+1} donc si $\sum_{k=n}^N (S_k - X_k)$ est multiple de Q_n

alors $\sum_{k=n+1}^N (S_k - X_k)$ est multiple de Q_{n+1} et si au même moment X_N décroît alors

$(Q_n m'_n + F_n)$ et $(Q_{n+1} m'_{n+1} + F_{n+1})$ augmentent simultanément. Ainsi, chaque fois qu'une autorisation de production arrive sur un étage n , elle est due à l'arrivée d'une autorisation de production sur un étage $n+1$. Par itération on montre qu'une autorisation de production à l'étage n implique l'arrivée simultanée d'autorisation de production sur chaque étage i pour i allant de $n+1$ à N .

Il est possible d'adapter d'autres politiques à une production par lots pour obtenir des équations similaires à l'équation (4.37), comme nous le montrons dans le paragraphe suivant pour le kanban.

4.5.3 Calcul de F_i pour une politique Kanban avec production par lots

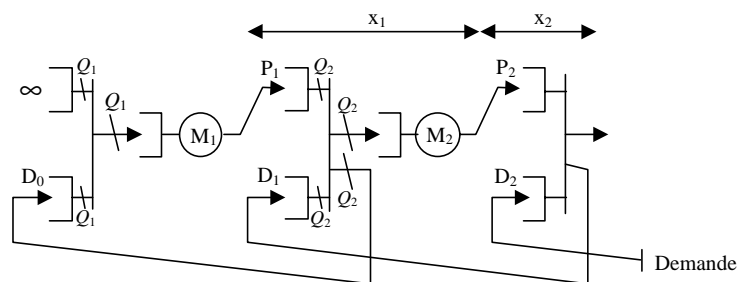


Figure 4.14 : Kanban pour une ligne de deux étages

En cas de production par lots, on suppose que pour tout n , on a toujours Q_n et P_n multiples de Q_{n+1} mais aussi F_n multiple de Q_n . Les mécanismes du kanban avec production par lots sont ceux du réseau de files d'attente de la figure 2.4 auxquels une contrainte est ajoutée sur le flux de pièces qui entre dans le processus de fabrication. En effet le nombre de pièces qui entrent dans F_n doit être un multiple de Q_n . Ainsi la station de synchronisation qui relie P_n et D_n ne laissera passer que des quantités multiples de Q_{n+1} . Un réseau de files d'attente sur la figure 4.14 modélise ce système. Comme pour le kanban modifié il existe un blocage qui modifie le

comportement de la politique. Ce type de blocage existe quand D_{i-1} est inférieur à Q_i , la taille des lots entrant dans F_i . L'équation (4.16) devient alors :

$$\begin{aligned} v_1(t) &= E_{Q_1} [v_2(t)] \\ v_i(t) &= E_{Q_i} [K_{i-1} \otimes u_{i-1}(t) \oplus v_{i+1}(t)] \quad \text{pour } 1 < i < N + 1 \\ v_{N+1}(t) &= K_N \otimes u_N(t) \oplus d(t) \end{aligned} \quad (4.39)$$

Avec les propriétés suivantes :

$$E_Q [A \oplus B] = E_Q [A] \oplus E_Q [B]$$

$$E_{Q_i} [E_{Q_j} [B]] = E_{Q_j} [E_{Q_i} [B]] = E_{Q_i} [B] \quad \text{pour } i < j$$

on trouve par des remplacements successifs une expression de $v_i(t)$ en fonction de $d(t)$ et de $u_i(t)$:

$$v_i(t) = E_{Q_i} \left[\bigoplus_{k=i}^{N+1} (K_{k-1} \otimes u_{k-1}(t)) \oplus d(t) \right] \quad \text{pour } i > 1 \quad (4.40)$$

Normalement $u_i(0) = 0$ donc $u_i(t)$ est multiple de Q_i , on a aussi $F_i(0) = 0$ et en utilisant les formules (4.10), (4.39) et (4.40) on trouve :

$$F_i(t) = E_{Q_i} \left[\bigoplus_{k=i}^{N+1} (K_{k-1} \otimes u_{k-1}(t) / u_i(t)) \oplus d(t) / u_i(t) \right] \quad \text{pour } i > 1$$

$$F_1(t) = E_{Q_1} \left[\bigoplus_{k=2}^{N+1} (K_{k-1} \otimes u_{k-1}(t) / u_1(t)) \oplus d(t) / u_1(t) \right]$$

Puis avec (4.12) et sachant que $X_i(0) = K_i$, on obtient :

$$F_i(t) = E_{Q_i} \left[X_{i-1}(t) \oplus K_i(t) \bigoplus_{k=i+1}^N \left(\bigotimes_{l=i}^k K_l / \bigotimes_{l=i}^{k-1} X_l(t) \right) \oplus \bigotimes_{k=i}^N K_k / \bigotimes_{k=i}^N X_k(t) \right] \quad \text{pour } i > 1$$

$$F_1(t) = E_{Q_1} \left[K_1(t) \bigoplus_{k=2}^N \left(\bigotimes_{l=1}^k K_l / \bigotimes_{l=1}^{k-1} X_l(t) \right) \oplus \bigotimes_{k=1}^N K_k / \bigotimes_{k=1}^N X_k(t) \right]$$

On a exprimé F_i en fonction de X et la formule obtenue est très proche de (4.19) la formule du kanban sans production par lots. Il est judicieux de choisir K_i multiple de Q_{i+1} car ainsi X_i sera multiple de Q_{i+1} , ce qui empêche l'existence de stocks résiduels non utilisables.

A chaque lot poussé dans F_i correspond une variation de $v_i(t)$ lorsqu'il y a suffisamment de pièces qui sont entrées en P_{i-1} . En (4.39) on peut observer que v_i est autorisé à augmenter suite à une augmentation de v_{i+1} lorsque $K_{i-1} + u_{i-1} > v_{i+1}$. Ainsi, un peu comme avec la politique (Q, r^i) , il y a synchronisation de la production des lots entre deux étages successifs.

Si l'on souhaite avoir un système kanban produisant par lots, plus général car n'ayant pas ce type de synchronisation entre les étages dans la prise en charge des lots au niveau des stations M_i , il suffit de prendre un système kanban identique à celui de la figure 2.4 et de définir chaque station M_i d'un étage i comme une station produisant des lots de taille Q_i .

Il n'y a aucun phénomène de blocage à l'entrée de ce type de processus de fabrication, donc la formule pour F_i reste la même qu'en (4.19). Ce qui change c'est le calcul du nombre de pièces servies par M_i :

$$M_i = Q_i \oplus E_{Q_i} [F_i] \quad (4.41)$$

Puisque F_i a été utilisée sans modification il n'y a pas de contrainte sur la valeur du paramétrage, ceci autorise le décalage des ordres de fabrication entre les étages.

Cette technique de construction d'une politique kanban produisant par lots à partir d'une politique kanban ne produisant pas par lots peut être adaptée à d'autres politiques ne produisant pas par lots et dont la valeur F_i est fonction de X . On obtient alors la quantité de pièces en cours de service dans les processus de fabrication par la formule (4.41). On peut cependant remarquer qu'il est toujours nécessaire d'avoir des composantes X_i multiples de Q_{i+1} pour éviter la présence de pièces inutiles dans les stocks.

4.6 Application aux systèmes de production avec assemblage

Considérons maintenant un système de production comportant un atelier d'assemblage. En amont de cet atelier, il y a deux lignes qui produisent deux produits intermédiaires différents.

Ces deux types de sous-produits entrent simultanément dans l'atelier d'assemblage. Cette simultanéité peut être obtenue par l'utilisation d'une station de synchronisation commune aux stocks de produits intermédiaires à assembler. La sortie de cette station de synchronisation est ensuite dirigée vers le processus de fabrication qui effectue l'assemblage.

Pour gérer ce type de système il peut être nécessaire de transmettre des informations vers les lignes de production qui se trouvent en amont de chaque station d'assemblage. Suivant les politiques, cette transmission diffère comme cela peut être visible sur les figures 2.16 et 2.17 pages 28 et 29 du chapitre 2 qui représentent respectivement le basestock et le kanban en cas d'assemblage.

Pour définir X le vecteur d'état global du système, il faut s'adapter à la structure particulière d'un système de production avec assemblage. Pour des étages situés en aval de la station de synchronisation on définit les variables suivantes :

$$X_i = F_{i+1} + P_i \text{ et } X_N = P_N - D_N \quad (4.42)$$

On considère l'assemblage de deux produits. La production de ces produits s'effectue dans deux portions de ligne nommées ligne a et ligne b . Il y a ensuite assemblage à l'étage n_A . Les variables d'état global pour ces deux portions de ligne sont les mêmes qu'en (4.42) mais des exposants indiquent s'il s'agit de la ligne a ou b :

$$X_i^a = F_{i+1}^a + P_i^a \text{ et } X_i^b = F_{i+1}^b + P_i^b \quad (4.43)$$

Au niveau de la station d'assemblage les variables d'état global sont les suivantes :

$$X_{n_A}^a = F_{n_A+1} + P_{n_A}^a \quad \text{et} \quad X_{n_A}^b = F_{n_A+1} + P_{n_A}^b \quad (4.44)$$

La figure 4.15 illustre chacune de ces variables d'état global.

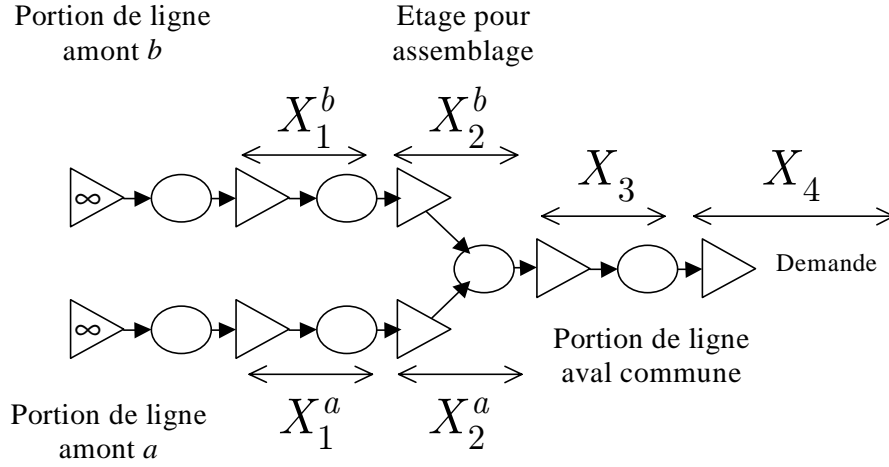


Figure 4.15 : Variables d'état global en cas d'assemblage de deux lignes de production

Pour simplifier l'écriture des calculs, les composantes X_i de la portion de ligne aval pourront être notées X_i^a ou X_i^b (on a donc $X_i = X_i^a = X_i^b$). Par exemple sur la figure 4.15 au lieu de parler de X_3 et X_4 on pourra parler de X_3^a et X_4^a , ou X_3^b et X_4^b , avec $X_3 = X_3^a = X_3^b$ et $X_4 = X_4^a = X_4^b$. Le vecteur d'état X étant défini, on peut à présent déterminer F_i en fonction de ce dernier pour différentes politiques.

4.6.1 Calcul de F_i pour un basestock avec assemblage

Les calculs s'effectuent comme pour obtenir le résultat (4.15) avec cependant quelques petites modifications au niveau de la station d'assemblage. Avec les notations de l'algèbre $(\min,+)$, on commence par calculer v_i :

$$\begin{aligned} v_i(t) &= S_{i-1} \otimes u_{i-1}(t) \oplus d(t) && \text{pour } i > n_A \\ v_{n_A}(t) &= S_{n_A-1}^a \otimes u_{n_A-1}^a(t) \oplus S_{n_A-1}^b \otimes u_{n_A-1}^b(t) \oplus d(t) \\ v_i^a(t) &= S_{i-1}^a \otimes u_{i-1}^a(t) \oplus d(t) && \text{pour } 1 < i < n_A \\ v_1^a(t) &= d(t) \end{aligned}$$

Sachant que $F_i(0) = 0$ et en utilisant la formule (4.10), on trouve :

$$\begin{aligned} F_i(t) &= S_{i-1} \otimes u_{i-1}(t)/u_i(t) \oplus d(t)/u_i(t) && \text{pour } i > n_A \\ F_{n_A}(t) &= S_{n_A-1}^a \otimes u_{n_A-1}^a(t)/u_{n_A}(t) \oplus S_{n_A-1}^b \otimes u_{n_A-1}^b(t)/u_{n_A}(t) \oplus d(t)/u_{n_A}(t) \\ F_i^a(t) &= S_{i-1}^a \otimes u_{i-1}^a(t)/u_i(t) \oplus d(t)/u_i(t) && \text{pour } 1 < i < n_A \\ F_1^a(t) &= d(t)/u_1^a(t) \end{aligned} \quad (4.45)$$

En notant S_i^a ou S_i^b , pour $n_A \leq i \leq N$, les constantes S_i de la portion de ligne aval (on a donc $S_i = S_i^a = S_i^b$), alors pour $1 \leq i \leq N$, on a $X_i^a(0) = S_i^a$.

Avec (4.12), et sachant que $X_i^a(0) = S_i^a$, on obtient :

$$\begin{aligned}
F_i^a(t) &= X_{i-1}^a(t) \oplus \bigotimes_{k=i}^N S_k^a / \bigotimes_{k=i}^N X_k^a(t) \quad \text{pour } 1 < i < n_A \text{ ou } n_A < i \leq N \\
F_{n_A}(t) &= X_{n_A-1}^a(t) \oplus X_{n_A-1}^b(t) \oplus \bigotimes_{k=n_A}^N S_k^a / \bigotimes_{k=n_A}^N X_k^a(t) \\
F_1^a(t) &= \bigotimes_{k=1}^N S_k^a / \bigotimes_{k=1}^N X_k^a(t)
\end{aligned} \tag{4.46}$$

On a une formule de la valeur de F_i exprimée en fonction de X et des paramètres S_i du basestock. Cette formulation est très proche d'une forme dite canonique.

4.6.2 Calcul de F_i pour un kanban avec assemblage à libération simultanée

On commence par calculer v_i comme cela a été fait en (4.16) :

$$\begin{aligned}
v_{N+1}(t) &= K_N \otimes u_N(t) \oplus d(t) \\
v_i(t) &= K_{i-1} \otimes u_{i-1}(t) \oplus v_{i+1}(t) \quad \text{pour } N+1 > i > n_A \\
v_{n_A}(t) &= K_{n_A-1}^a \otimes u_{n_A-1}^a(t) \oplus K_{n_A-1}^b \otimes u_{n_A-1}^b(t) \oplus v_{n_A+1}(t) \\
v_i^a(t) &= K_{i-1}^a \otimes u_{i-1}^a(t) \oplus v_{n_A+1}^a(t) \quad \text{pour } 1 < i < n_A \\
v_1^a(t) &= v_2^a(t)
\end{aligned}$$

Par des remplacements successifs, on exprime les compteurs v par des compteurs u et d :

Pour $N+1 \geq i > n_A$, on a $v_i(t) = \bigoplus_{k=i-1}^N (K_k \otimes u_k(t)) \oplus d(t)$ et

$$v_{n_A}(t) = K_{n_A-1}^a \otimes u_{n_A-1}^a(t) \oplus K_{n_A-1}^b \otimes u_{n_A-1}^b(t) \oplus \bigoplus_{k=n_A}^N (K_k \otimes u_k(t)) \oplus d(t)$$

Pour $1 < i < n_A$, on a :

$$v_i^a(t) = \bigoplus_{k=i-1}^{n_A-1} (K_k^a \otimes u_k^a(t)) \oplus K_{n_A-1}^b \otimes u_{n_A-1}^b(t) \oplus \bigoplus_{k=n_A}^N (K_k \otimes u_k(t)) \oplus d(t)$$

et $v_1^a(t) = v_2^a(t)$.

En notant K_i par K_i^a , sachant que $F_i(0) = 0$, $X_i^a(0) = K_i^a$ et en utilisant les formules

(4.10) et (4.12), on trouve :

Pour $N+1 \geq i > n_A$ on a

$$F_i(t) = X_{i-1}^a(t) \oplus K_i^a \oplus \bigoplus_{j=i}^{N-1} \left(\bigotimes_{k=i}^{j+1} K_k^a / \bigotimes_{k=i}^j X_k^a(t) \right) \oplus \bigotimes_{k=i}^N K_k^a / \bigotimes_{k=i}^N X_k^a(t),$$

$$F_{n_A}(t) = X_{n_A-1}^a(t) \oplus X_{n_A-1}^b(t) \oplus K_{n_A}^a \oplus \bigoplus_{j=n_A}^{N-1} \left(\bigotimes_{k=n_A}^{j+1} K_k^a / \bigotimes_{k=n_A}^j X_k^a(t) \right) \oplus \bigotimes_{k=n_A}^N K_k^a / \bigotimes_{k=n_A}^N X_k^a(t)$$

Pour $i < n_A$ et en posant $X_0^a(t) = \varepsilon$, on a

$$F_i^a(t) = X_{i-1}^a(t) \oplus K_i^a \oplus \left(\bigoplus_{j=i+1}^N \left(\bigotimes_{k=i}^j K_k^a / \bigotimes_{k=i}^{j-1} X_k^a(t) \right) \right) \oplus \bigotimes_{k=i}^N K_k^a / \bigotimes_{k=i}^N X_k^a(t) \oplus X_{n_A-1}^b(t) \otimes \bigotimes_{k=i}^{n_A-1} K_k^a / \bigotimes_{k=i}^{n_A-1} X_k^a(t)$$

On a une formule de la valeur de F_i exprimée en fonction de X et des constantes du kanban.

On peut écrire ces formules avec des opérateurs usuels, la structure des formules est alors plus facile à percevoir :

Pour $N + 1 \geq i > n_A$ on a

$$F_i = \min(\begin{array}{cccccccc} -X_N^a & \cdots & \cdots & -X_i^a & +K_N^a & \cdots & \cdots & +K_i^a \\ & & \ddots & -X_i^a & +K_N^a & & & +K_i^a \\ & & & \ddots & -X_i^a & & \ddots & +K_i^a \\ & & & & -X_i^a & & & +K_{i+1}^a & +K_i^a \\ & & & & & & & & K_i^a \end{array} , X_{i-1}^a)$$

et

$$F_{n_A} = \min(\begin{array}{cccccccc} -X_N^a & \cdots & \cdots & -X_{n_A}^a & +K_N^a & \cdots & \cdots & +K_{n_A}^a \\ & & \ddots & -X_{n_A}^a & +K_N^a & & & +K_{n_A}^a \\ & & & \ddots & -X_{n_A}^a & & \ddots & +K_{n_A}^a \\ & & & & -X_{n_A}^a & & & +K_{n_A+1}^a & +K_{n_A}^a \\ & & & & & & & & K_{n_A}^a \end{array} , X_{n_A-1}^a , X_{n_A-1}^b)$$

Pour $i < n_A$ et en posant $X_0^a = \infty$ on a

$$F_i^a = \min(\begin{array}{cccccccc} -X_N^a - X_{N-1}^a - \cdots - X_i^a & +K_N^a + & \cdots & +K_i^a \\ -X_{N-1}^a - \cdots - X_i^a & +K_N^a + & \cdots & +K_i^a \\ \ddots & -X_i^a & \ddots & +K_i^a \\ -X_i^a & & & K_{i+1}^a + K_i^a \\ & & & K_i^a \end{array} , X_{i-1}^a , (X_{n_A-1}^b - X_{n_A-1}^a) - X_{n_A-2}^a - \cdots - X_i^a , K_{n_A-1}^a + \cdots + K_i^a)$$

Ces résultats sont assez proches de ceux obtenus en (4.19) pour le kanban. Les formules des F_i sont identiques pour $i < n_A$, mais pour F_{n_A} et F_i^a des différences proviennent de termes prenant en considération le nombre de pièces présentes en $P_{n_A-1}^b$. Pour F_{n_A} les termes $X_{n_A-1}^a$ et $X_{n_A-1}^b$ permettent de contrôler que les pièces à assembler existent. Pour F_i^a le terme

utilisant la valeur $(X_{n_A-1}^b - X_{n_A-1}^a)$ est celui qui exprime les phénomènes de blocages causés par une valeur de $P_{n_A-1}^b$ inférieure à $P_{n_A-1}^a$.

Des résultats similaires peuvent être obtenus pour les politiques kanban étendu et kanban généralisé dans un système avec assemblage.

4.7 Application au MRP

La politique MRP (cf. 2.5.6) que nous considérons est issue du modèle PAC de Buzacott [BUZ 1993]. On peut la représenter par un réseau de files d'attente similaire à celui d'une politique basestock mais ayant des temporisations pour retarder la transformation des demandes prévisionnelles en ordre de fabrication.

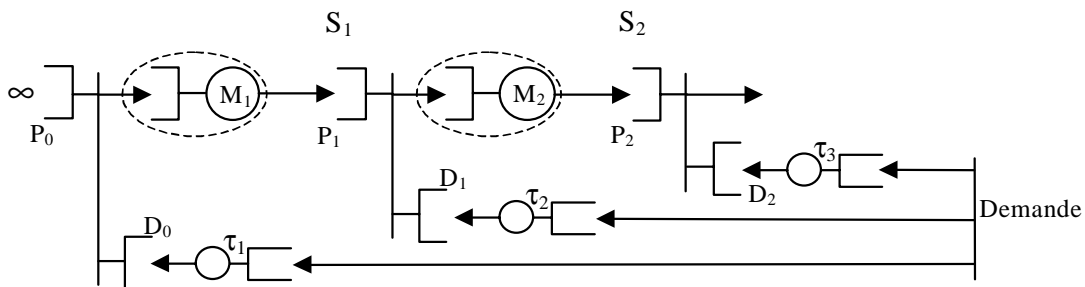


Figure 4.16 : MRP représenté par réseau de files d'attente avec temporisations.

Les files d'attente D_0 , D_1 et D_2 sont précédées de temporisations constantes respectives τ_1 , τ_2 et τ_3 . Notre formulation doit être modifiée pour pouvoir modéliser ces temporisations. Le calcul de F_i est le suivant :

$$\begin{aligned}
 F_i(t) &= v_i(t) - u_i(t) \\
 &= \min(d(t - \tau_i) - u_i(t), u_{i-1}(t) + S_{i-1} - u_i(t)) \\
 &= \min(d(t - \tau_i) - d(t) + d(t) - u_i(t), X_{i-1}(t)) \\
 &= \min\left(d(t - \tau_i) - d(t) + \sum_{j=i}^N S_j - \sum_{j=i}^N X_j(t), X_{i-1}(t)\right)
 \end{aligned}$$

Si le nombre de demandes parvenues pendant les dernières τ_i unités de temps est noté ΔD_i , alors avec ΔD le vecteur dont les composantes sont ΔD_i , on peut exprimer F_i de la sorte :

$$F_i(X, \Delta D) = \min\left(X_{i-1}, \sum_{j=i}^N S_j - \sum_{j=i}^N X_j - \Delta D_i\right) \quad (4.47)$$

A chaque instant il faut connaître le nombre de produits présents dans la ligne et le nombre de demandes prévisionnelles faites pendant les dernières τ_i unités de temps.

Supposons que les demandes prévisionnelles portent sur une période de longueur τ_3 , alors ΔD_i représente le nombre de demandes fermes prévues entre les instants $\tau_3 - \tau_i$ et τ_3 .

Karaesmen, Buzacott et Dallery [KAR 1999] ont proposé un modèle de basestock avec temporisation des prévisions variant suivant l'horizon de prévision. Dans ce cas où la temporisation τ_i varierait selon une certaine loi de probabilité, il n'est plus possible a priori d'utiliser la formule (4.47). Ainsi, les politiques à flux poussé semblent peu adaptées à la formulation des politiques que nous proposons.

4.8 Conclusions

Dans la plupart des cas, pour un système à flux tiré, une méthode de calcul proche de celle donnée par l'algorithme 1 (cf. 4.3) permet d'obtenir la valeur de F_i exprimée en fonction de X . Nous avons traité le cas de différents types de processus de fabrication, de systèmes avec production par lots ou ayant une station d'assemblage, à chaque fois le calcul a été possible.

Il est intéressant d'observer que les formules obtenues sont toutes assez proches d'une formulation (4.30) qualifiée de canonique. Quand l'algorithme 1 est utilisable, on obtient cette formulation.

Pour une production par lots, on trouve, suivant les cas, soit la formule (4.41), soit une formulation ayant l'aspect suivant :

$$F_1(t) = E_{Q_1} \left[\bigoplus_{j=0}^N \left(C_{(1,j)} / \bigotimes_{k=1}^j X_k(t) \right) \right]$$

$$F_i(t) = E_{Q_i} \left[\bigoplus_{j=i-1}^N \left(C_{(i,j)} / \bigotimes_{k=i}^j X_k(t) \right) \oplus X_{i-1} \right] \quad \text{pour } i > 1$$

De même pour un système avec assemblage, on obtient aussi une formulation qui, à quelques termes près, est similaire à (4.30). Avec ce type de formulation, que nous avons qualifiée de canonique, les politiques usuelles ne diffèrent que par quelques constantes. Lorsque pour deux politiques ces constantes sont identiques, on peut alors affirmer que ces politiques ont une dynamique identique.

Ce type de comparaisons est l'objet du chapitre suivant. Il s'agira de trouver toute identité de comportement possible entre les politiques basestock, kanban, kanban étendu et kanban généralisé, plusieurs types de processus de fabrication sont considérés.

Chapitre 5

Recherche des identités de fonctionnement entre politiques

Nous allons utiliser les formulations précédemment décrites pour rechercher des identités de comportement entre plusieurs politiques. On considère que deux systèmes ayant un même processus de fabrication ont un comportement identique s'ils ont une même dynamique. Dans le cas d'un processus de fabrication général ne traitant pas les pièces par lots (le cas de lignes produisant par lots n'a pas été considéré) et avec des critères de type (a) ou (b), les dynamiques sont respectivement définies par les définitions B.1.a ou B.1.b de l'annexe B.

Si les processus de fabrication sont constitués d'un stock tampon suivi d'un monoserveur, on pourra utiliser, avec des critères de type (a) ou (b), une dynamique respectivement définie par les définitions 3.3.a ou 3.3.b des sections 3.2.2 et 3.2.3. Dans le cas général pour des processus de fabrication ne traitant pas les pièces par lots (le cas de lignes produisant par lots n'a pas été considéré) les dynamiques sont définies suivant le contexte par les définitions B.1.a ou B.1.b de l'annexe B.

Pour un critère de type (a), on s'intéresse à des processus de fabrication F_i composés d'un stock tampon suivi d'un mécanisme de production général (une station monoserveur, multiserveur ou serveur infini ; plusieurs stations en série, ...) d'où les pièces sortent une à une. Il est alors possible d'utiliser le corollaire 3.2.a de la section 3.3.3.1. Ce dernier permet de déterminer la dynamique au sens de la définition B.1.a en utilisant pour chaque étage la valeur de F_i en fonction de X . On peut dès lors savoir si deux systèmes 1 et 2 ont un comportement identique pour un critère de type (a). Il faut pour cela regarder si les fonctions $F_i^1(X)$ et $F_i^2(X)$ qui définissent respectivement F_i en fonction de X sur ces deux systèmes sont identiques pour tout i .

Pour un critère de type (b), on s'intéresse au cas de processus de fabrication composés d'un stock tampon suivi d'un serveur infini, d'un multiserveur ou d'un monoserveur. (Rappelons que l'on veut utiliser le corollaire 3.2.b de l'annexe B section B.V alors les processus de fabrication composés d'un stock tampon et suivis d'un mécanisme de production général ne peuvent pas être considérés). Le corollaire 3.2.b nous permet pour les cas considérés de

définir la dynamique selon la parenthèses définition B.1.b à partir de la valeur de M_i fonction de X . On peut dès lors savoir si deux systèmes 1 et 2 ont un comportement identique pour un critère de type (b). Il faut pour cela regarder si les fonctions $M_i^1(X)$ et $M_i^2(X)$ qui respectivement définissent M_i en fonction de X sur ces deux systèmes, sont identiques pour tout i . On peut calculer la valeur de $M_i(X)$ à partir de la valeur de $F_i(X)$. Pour un stock tampon suivi respectivement d'un serveur infini, d'un multiserveur ayant n_i serveurs ou d'un monoserveur, on a respectivement $M_i = F_i$, $M_i = \min(n_i, F_i)$ ou $M_i = \min(1, F_i)$.

Ainsi, il est suffisant de montrer que les fonctions $F_i(X)$ sont identiques pour montrer que deux systèmes ont un comportement identique pour un critère de type (a) ou de type (b).

Dans les cas énoncés ci-dessus, l'égalité entre les valeurs F_i (respectivement M_i) fonctions de X est suffisante pour montrer l'égalité des dynamiques pour un critère de type (a) (respectivement de type (b)). Cependant, est-il nécessaire d'avoir des fonctions $F_i(X)$ (respectivement $M_i(X)$) identiques pour avoir des dynamiques identiques ? Supposons que deux systèmes 1 et 2 aient des valeurs F_i (respectivement M_i) égales à $F_i^1(X)$ et $F_i^2(X)$ (respectivement $M_i^1(X)$ et $M_i^2(X)$). Si $F_i^1(X) \neq F_i^2(X)$ alors en utilisant le lemme 3.4.a, on a $E = \mathfrak{E}[E_X]$ et, puisque $\mathfrak{E}(X)$ utilise $F_i(X)$, les ensembles E des systèmes 1 et 2 ne peuvent pas être égaux (respectivement, si $M_i^1(X) \neq M_i^2(X)$ alors en utilisant le lemme 3.4.b, on a $E' = \mathfrak{E}'[E'_X]$ et puisque $\mathfrak{E}'(X)$ utilise $M_i(X)$ les ensembles E' des systèmes 1 et 2 ne peuvent pas être égaux). Si les ensembles E (respectivement E' pour un critère de type (b)) définissant les états admissibles sont différents dans les deux systèmes alors les dynamiques sont également différentes. Il est donc nécessaire d'avoir des fonctions $F_i(X)$ (respectivement $M_i(X)$) identiques pour avoir des dynamiques identiques.

Pour les cas cités, avoir des fonctions $F_i(X)$ (ou $M_i(X)$) identiques est donc une condition nécessaire et suffisante pour montrer que deux politiques ont un comportement identique. Dans les sections qui suivent nous allons principalement examiner dans quels cas les politiques basestock, kanban, kanban étendu ou kanban généralisé, peuvent avoir un comportement identique.

5.1 Quelques équivalences de comportement connues

Il existe plusieurs équivalences de comportement connues pour les politiques basestock, kanban, kanban étendu et kanban généralisé que nous avons choisi d'étudier [BUZ 1993] [DAL 1995] :

- On sait qu'une politique kanban étendu est équivalente à une politique basestock si le nombre de ses kanbans sur chaque étage tend vers l'infini. En effet la remontée de la demande ne peut alors plus être arrêtée par une absence de kanbans disponibles.
- Le kanban étendu a un comportement équivalent à la politique kanban lorsque sur chacun des étages le nombre de kanbans est égal au niveau nominal de pièces attendues dans le stock de cet étage.
- Pour le kanban généralisé, on sait qu'il est équivalent à une politique basestock si le nombre de ses kanbans sur chaque étage tend vers l'infini. Comme pour le kanban étendu ce nombre illimité de kanbans permet une remontée immédiate des demandes sur toute la ligne.
- Le kanban généralisé a aussi un comportement équivalent à la politique kanban lorsque sur chacun des étages le nombre de kanbans est égal au niveau nominal de pièces attendues dans le stock de cet étage.

On peut remarquer que l'identité de comportement dépend principalement du paramétrage particulier attribué à une politique. Certaines équivalences sont dues au comportement limite des politiques lorsque le paramétrage tend vers l'infini. Parmi ces politiques étudiées, il existe d'autres identités de comportement, ainsi dans les sections qui suivent, nous avons effectué cette recherche de façon exhaustive par l'emploi des formulations précédemment calculées.

5.2 Pour une ligne de deux étages, comparaison entre kanban étendu et kanban généralisé

Pour une ligne de deux étages, les formules définissant le nombre de produits présents dans les processus de fabrication, ayant un stock tampon, sont les suivantes :

- On a pour le kanban étendu :

$$F_1^e(X) = \min(K_1^e, K_1^e + K_2^e - X_1, S_1^e + S_2^e - X_1 - X_2)$$

$$F_2^e(X) = \min(K_2^e, X_1, S_2^e - X_2)$$

- On a pour le kanban généralisé :

$$F_1^g(X) = \min(K_1^g, S_1^g + K_2^g - X_1, S_1^g + S_2^g - X_1 - X_2)$$

$$F_2^g(X) = \min(K_2^g, X_1, S_2^g - X_2)$$

Les exposants e et g indiquent respectivement qu'une valeur, ou un paramètre, appartient à une politique kanban étendu ou kanban généralisé.

L'identité des dynamiques va dépendre du paramétrage de ces politiques mais aussi du type de processus de fabrication choisi. Nous traitons le cas des processus de fabrication avec stock tampon suivi d'un monoserveur, et le cas plus général où les processus de fabrication

sont quelconques mais avec un stock tampon à leur entrée et des pièces qui ne sortent pas par lots de plusieurs pièces.

5.2.1 Processus de fabrication quelconque précédé d'un stock tampon et tel que les pièces sortent une à une.

Dans ce cas, quel que soit le type de critère utilisé, il y a un même comportement des systèmes si, et seulement si, les fonctions $F_i^g(X)$ et $F_i^e(X)$ sont identiques pour tout X admissible, avec $i = 1$ et $i = 2$. Ainsi on a, pour tout X admissible :

$$\begin{aligned} & \min(K_1^g, S_1^g + K_2^g - X_1, S_1^g + S_2^g - X_1 - X_2) \\ &= \min(K_1^e, K_1^e + K_2^e - X_1, S_1^e + S_2^e - X_1 - X_2) \end{aligned}$$

et

$$\min(K_2^g, X_1, S_2^g - X_2) = \min(K_2^e, X_1, S_2^e - X_2)$$

Pour que X soit un vecteur d'état global admissible, on vérifie qu'il appartient au polyèdre défini en (4.31). En dehors de ce polyèdre, X n'est pas admissible car le nombre de pièces présentes dans certains processus de fabrication est négatif.

Chaque équation qu'il faut vérifier est composée de fonctions \min dont les arguments sont des fonctions affines de X . On va commencer par vérifier s'il n'y a pas d'arguments inutiles dans ces fonctions \min . Pour qu'un argument soit inutile dans une fonction \min , il est nécessaire et suffisant qu'il soit supérieur ou égal aux autres arguments pour tout état global admissible et que sa suppression ne crée pas d'état global admissible supplémentaire.

Les fonctions \min comparées sont d'une forme identique à (4.30) qui est dite canonique. On compare donc deux fonctions de la forme $\min(C_{(1,0)}, C_{(1,2)} - X_1 - X_2, C_{(1,1)} - X_1)$ et deux fonctions de la forme $\min(X_1, C_{(2,1)}, C_{(2,2)} - X_2)$. Pour faciliter la comparaison de ces fonctions, on supprime les arguments inutiles.

Les deux fonctions \min qui sont comparées ont des arguments composés d'une partie linéaire additionnée d'une constante. Les arguments d'une fonction \min ont des parties linéaires deux à deux distinctes mais entre les deux fonctions \min comparées les arguments ont des parties linéaires identiques. Les arguments inutiles supprimés doivent avoir une même partie linéaire. Pour montrer qu'il y a identité entre les deux fonctions il reste alors à vérifier que les constantes $C_{(i,j)}$ qui s'ajoutent à la partie linéaire sont égales de part et d'autre.

Pour le cas considéré entre kanban étendu et kanban généralisé on montre que tous les arguments sont utiles sous certaines conditions :

- ♦ Avec $X = (0, -K_1^g - 1)$, on trouve $F_1^g = \min(K_1^g, S_1^g + K_2^g) > 0$ et $F_2^g = 0$, donc ce vecteur d'état global X est toujours admissible, et l'argument K_1^g est utile pour $K_1^g < S_1^g + K_2^g$. Si $K_1^g \geq S_1^g + K_2^g$ alors K_1^g ne peut pas être utile même en

modifiant X , car pour que K_1^g devienne le plus petit argument de la fonction \min il faudrait faire décroître X_1 , or X_1 est nul et ne doit pas devenir négatif. Donc l'argument K_1^g est utile si, et seulement si, $K_1^g < S_1^g + K_2^g$.

- ◆ Avec $X = (0, S_1^e + S_2^e - K_1^e - 1)$, on trouve $F_1^e = K_1^e$ et $F_2^e \geq 0$ donc l'argument K_1^e est utile.
- ◆ Avec $X = (S_1^g + K_2^g, S_2^g - K_2^g - 1)$, on montre que l'argument $S_1^g + K_2^g - X_1$ est utile.
- ◆ Avec $X = (K_1^e + K_2^e, S_1^e + S_2^e - K_1^e - K_2^e - 1)$, on montre que l'argument $K_1^e + K_2^e - X_1$ est utile.
- ◆ Avec $X = (S_1^g + K_2^g - 1, S_2^g - K_2^g)$, on montre que l'argument $S_1^g + S_2^g - X_1 - X_2$ est utile.
- ◆ Avec $X = (K_1^e + K_2^e - 1, S_1^e + S_2^e - K_1^e - K_2^e)$, on montre l'utilité de l'argument $S_1^e + S_2^e - X_1 - X_2$.
- ◆ Avec $X = (K_2^g + 1, S_2^g - K_2^g - 1)$, on trouve $F_2^g = K_2^g$ et $F_1^g = \min(K_1^g, S_1^g - 1)$, on montre alors que l'argument K_2^g est utile si, et seulement si, $S_1^g \geq 1$.
- ◆ Avec $X = (K_2^e + 1, S_2^e - K_2^e - 1)$, on trouve $F_2^e = K_2^e$ et $F_1^e \geq 0$ donc l'argument K_2^g est utile.

On suppose que tous les arguments sont utiles (cela revient à supposer $S_1^g \geq 1$ et $K_1^g < S_1^g + K_2^g$) alors pour montrer l'égalité des fonctions \min , pour tout X admissible, il est suffisant (et nécessaire, cela peut se montrer en ramenant ce problème à une comparaison de polyèdres en nombre entier) de résoudre :

$$\begin{cases} K_1^g = K_1^e \\ S_1^g + K_2^g - X_1 = K_1^e + K_2^e - X_1 \\ S_1^g + S_2^g - X_1 - X_2 = S_1^e + S_2^e - X_1 - X_2 \text{ avec } S_1^e \leq K_1^e \text{ et } S_2^e \leq K_2^e \\ K_2^g = K_2^e \\ S_2^g - X_2 = S_2^e - X_2 \end{cases}$$

on obtient alors :

$$\begin{cases} S_2^g = S_2^e \\ S_1^g = S_1^e \\ K_2^g = K_2^e \text{ avec } S_1^e \leq K_1^e \text{ et } S_2^e \leq K_2^e \\ K_1^g = K_1^e \\ K_1^e = S_1^e \end{cases} \quad (5.1)$$

Si (5.1) est vérifié alors on a $S_1^e \geq 1$ et $K_1^g < S_1^g + K_2^g$ (car $K_1^e \geq 1$, $S_1^g = K_1^e$ et $K_2^g \geq 1$), ce qui permet d'affirmer qu'aucun argument des fonctions \min n'est inutile.

La contrainte $K_1^e = S_1^e$ signifie qu'il n'est possible de trouver une équivalence entre les deux politiques que si le paramétrage rend la gestion de la maille amont équivalente à celle d'un simple kanban. La figure 5.1 illustre de gauche à droite par deux réseaux de files d'attente ce que représente la contrainte $K_1^e = S_1^e$ pour le kanban étendu et $K_1^g = S_1^g$ pour le kanban généralisé.

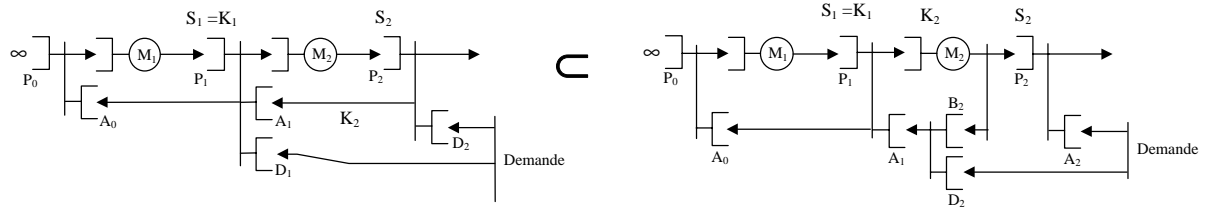


Figure 5.1 : L'ensemble des politiques kanban étendu vérifiant $K_1^e = S_1^e$ est inclus dans l'ensemble des politiques kanban généralisé vérifiant $K_1^g = S_1^g$.

Pour chaque politique kanban étendu vérifiant $K_1^e = S_1^e$, il est possible de définir une politique kanban généralisé équivalente en posant $S_2^g = S_2^e$, $S_1^g = S_1^e$, $K_2^g = K_2^e$ et $K_1^g = K_1^e$.

Par contre pour un système kanban généralisé vérifiant $K_1^g = S_1^g$, il n'est pas toujours possible de trouver un kanban étendu équivalent. Pour être possible, il est nécessaire d'avoir $K_2^g \geq S_2^g$, car sinon on ne peut avoir $K_2^e \geq S_2^e$ puisque $S_2^g = S_2^e$ et $K_2^g = K_2^e$.

Comme indiqué sur la figure 5.1, pour une ligne de deux mailles, l'ensemble des politiques kanban étendu vérifiant $K_1^e = S_1^e$ est donc inclus dans l'ensemble des politiques kanban généralisé vérifiant $K_1^g = S_1^g$.

5.2.2 Processus de fabrication avec monoserveur précédé d'un stock tampon.

On s'intéresse ici à un cas particulier de la section 5.2.1 où les processus de fabrication sont composés d'un stock tampon suivi d'une station monoserveur. On trouve dans [KAR 2000] des résultats similaires à ceux que nous présentons ci-dessous. Pour un critère de type *a* les systèmes considérés ont le même comportement si, et seulement si, les fonctions $F_i^g(X)$ et $F_i^e(X)$ sont identiques pour $i = 1$ et $i = 2$. Les résultats sont alors identiques à ceux obtenus précédemment pour un processus de fabrication avec stock tampon et pièces sortant une à une. Par contre pour un critère de type *b* le comportement des systèmes sera identique si, et seulement si, les valeurs $M_i^g(X)$ et $M_i^e(X)$ sont égales pour $i = 1$ et $i = 2$. Sachant que $M_i = \min(1, F_i)$, on a donc :

$$\begin{aligned} & \min(K_1^g, 1, S_1^g + K_2^g - X_1, S_1^g + S_2^g - X_1 - X_2) \\ &= \min(K_1^e, 1, K_1^e + K_2^e - X_1, S_1^e + S_2^e - X_1 - X_2) \end{aligned}$$

et

$$\begin{aligned} & \min(K_2^g, 1, X_1, S_2^g - X_2) \\ & = \min(K_2^e, 1, X_1, S_2^e - X_2) \end{aligned}$$

Puisque le nombre de kanbans est supérieur ou égal à un, on trouve après simplification :

$$S_2^g = S_2^e, S_1^g = S_1^e, K_2^g = K_1^e + K_2^e - S_1^e \text{ et } K_1^g \text{ quelconque.}$$

Quel que soit le paramétrage d'un kanban étendu, il est donc possible de définir tous les paramètres d'un kanban généralisé équivalent. En particulier avec $K_1^e \geq S_1^e$ et $K_2^e \geq 1$, on trouve $K_1^e + K_2^e - S_1^e \geq 1$, la valeur de K_2^g est donc bien définie puisqu'elle est supérieure ou égale à un.

Il n'est, par contre, pas toujours possible d'obtenir un système kanban étendu ayant le même comportement qu'un système kanban généralisé. Pour être équivalent, un système kanban étendu doit vérifier $S_2^e = S_2^g$, $S_1^e = S_1^g$, $K_1^e = S_1^e + K_2^g - K_2^e$, $K_1^e \geq S_1^e$ et $K_2^e \geq S_2^e$. De façon plus explicite on a le résultat équivalent suivant :

$$S_2^e = S_2^g, S_1^e = S_1^g, K_1^e = S_1^g + K_2^g - K_2^e \text{ et la contrainte } S_2^g \leq K_2^e \leq K_2^g.$$

Dans une ligne de deux mailles, avec un critère de type b , l'ensemble des politiques kanban étendu est donc inclus dans l'ensemble des politiques kanban généralisé.

Il est possible d'étudier les équivalences de comportement pour des processus de fabrication avec stock tampon et multiserveur. Les résultats obtenus sont alors un compromis de ceux trouvés pour le cas général et le cas monoserveur. Dans la section qui suit nous allons étendre, à des lignes de taille quelconque, les résultats obtenus pour des systèmes à deux mailles.

5.3 Kanban étendu et kanban généralisé sur une ligne de plusieurs étages

Dans un premier temps, nous considérons que les paramètres S_i et K_i sont finis, ainsi il n'est pas nécessaire d'étudier le comportement de ces deux politiques lorsque certains de ces paramètres tendent vers l'infini.

5.3.1 Comparaisons avec paramétrage à valeur finie

Après l'étude des lignes dont les processus de fabrication ont un stock tampon, nous traiterons le cas particulier du monoserveur.

5.3.1.1 Processus de fabrication avec stock tampon et pièces sortant une à une.

Les lignes considérées ici comportent N mailles, les processus de fabrication sont composés d'un stock tampon suivi d'un mécanisme de production quelconque composé d'une ou plusieurs stations et d'où les pièces sortent une à une. Pour le cas considéré, quel que soit le type de critère choisi, deux lignes de production auront un même comportement si, et seulement si, les valeurs $F_i^g(X)$ et $F_i^c(X)$ sont identiques pour i allant de 1 à N et pour chaque vecteur X admissible. On a donc

$$F_1^c(X) = \bigoplus_{k=2}^N \left(\bigotimes_{j=1}^{k-1} (K_j^c / X_j) \otimes K_k^c \right) \oplus \bigotimes_{k=1}^N (S_k^c / X_k) \oplus K_1^c$$

égal à

$$F_1^g(X) = \bigoplus_{j=2}^N \left(K_j^g \otimes \bigotimes_{k=1}^{j-1} (S_k^g / X_k) \right) \oplus \bigotimes_{k=1}^N (S_k^g / X_k) \oplus K_1^g$$

et pour $i > 1$ et $i \leq N$, on a

$$F_i^c(X) = \bigoplus_{k=i+1}^N \left(\bigotimes_{j=i}^{k-1} (K_j^c / X_j) \otimes K_k^c \right) \oplus \bigotimes_{k=i}^N (S_k^c / X_k) \oplus K_i^c \oplus X_{i-1}$$

égal à

$$F_i^g(X) = \bigoplus_{j=i+1}^N \left(K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k) \right) \oplus \bigotimes_{k=i}^N (S_k^g / X_k) \oplus K_i^g \oplus X_{i-1}$$

Chacune de ces formules est une fonction *min* dont les arguments sont des fonctions affines de X . On procède ici comme dans la section 5.2.1. On va commencer par vérifier qu'il n'y a pas d'arguments inutiles. Pour montrer qu'il y a identité entre les deux fonctions, il reste alors à vérifier que les constantes $C_{(i,j)}$ qui s'ajoutent à la partie linéaire sont égales de part et d'autre.

Examinons l'utilité de chaque argument des fonctions *min* utilisées dans ces formules dites canoniques. Il faut pour cela exhiber, avec chaque argument, un vecteur X admissible qui rend l'argument considéré strictement plus petit que les autres.

Examinons pour le kanban étendu, l'utilité de chaque argument des fonctions *min* utilisées dans $F_i^c(X)$.

Lemme 5.1 :

Pour le kanban étendu tous les arguments des fonctions *min* données en (4.23) pour définir $F_i^c(X)$ sont utiles, à l'exception de l'argument $\bigotimes_{k=i}^N (S_k^c / X_k)$ qui ne l'est pas si, et seulement si, $S_{i-1}^c = 0$.

□

Preuve :

- Relativement à la nécessité de considérer l'argument K_1^c , on peut vérifier aisément que le vecteur admissible $X = (-K_1^c - 1) \cdot 1_N$ permet de rendre tous les autres arguments strictement supérieurs.

- Pour l'argument K_i^e , on prend $X = (-K_i^e - 1) \cdot 1_N + (K_i^e + 1) \cdot 1_{i-1}$.

- Pour l'argument X_{i-1} , on prend $X = -1_N$.

- Pour l'argument $\left(\bigotimes_{j=i}^{k-1} (K_j^e / X_j) \otimes K_k^e \right)$, on prend

$$X = \sum_{\substack{j=1 \\ j \notin \{k-1, k\}}}^{N-1} (K_j^e \cdot 1_j) + (K_{k-1}^e + K_k^e) \cdot 1_{k-1} + \sum_{j=1}^N (S_j^e - K_j^e) \cdot 1_N.$$

- Pour l'argument $\bigotimes_{k=i}^N (S_k^e / X_k)$, on prend $X = S^e$ lorsque $S_{i-1}^e > 0$, sinon on peut

montrer par l'absurde que cet argument est inutile quand $S_{i-1}^e = 0$: Supposons qu'il

existe un vecteur X admissible tel que $\bigotimes_{k=i}^N (S_k^e / X_k)$ soit strictement le plus petit

argument de la fonction \min alors celui-ci est strictement inférieur à X_{k-1} et ainsi

$$\bigotimes_{k=i}^N (S_k^e / X_k) \otimes 0 / X_{i-1} = \bigotimes_{k=i-1}^N (S_k^e / X_k) < 0. X \text{ n'est donc pas admissible puisque}$$

$F_{i-1}^e(X) < 0$, il y a contradiction et ceci montre que l'argument $\bigotimes_{k=i}^N (S_k^e / X_k)$ est

inutile si $S_{i-1}^e = 0$. Cet argument correspond aux autorisations de production de type

basestock qui sont transmises instantanément à chaque arrivée d'une demande de

produits finis. □

Examinons à présent pour $F_i^g(X)$ les arguments de sa fonction \min :

Lemme 5.2 :

Pour le kanban généralisé, dans les fonctions \min données en (4.29) pour définir $F_i^g(X)$:

- X_{i-1} est toujours utile.

- K_N^g est utile si et seulement si $S_{N-1}^g > 0$.

- K_i^g pour $1 < i < N$ est utile ssi $S_{i-1}^g > 0$ et si $K_j^g + \sum_{k=i}^{j-1} S_k^g > K_i^g$ pour $j > i$.

- K_1^g est utile ssi $K_j^g + \sum_{k=1}^{j-1} S_k^g > K_1^g$ pour $j > 1$.

- $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ pour $i < j \leq N$ est utile ssi $S_{i-1}^g > 0$ et $K_l^g + \sum_{k=j}^{l-1} S_k^g > K_j^g$ avec

$$j < l.$$

- $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile ssi $S_{i-1}^g > 0$

□

Preuve :

- Pour montrer que l'argument K_N^g peut être strictement le plus petit de la fonction \min dans $F_N^g(X)$, on peut prendre le vecteur $X = (-K_N^g - 1) \cdot 1_N + (K_N^g + 1) \cdot 1_{N-1}$. Cependant ce vecteur n'est pas admissible si $1 > S_{N-1}^g = 0$ car alors $F_{N-1}^g(X)$ est négatif. Pour que l'argument K_N^g soit utile on doit avoir un vecteur admissible X tel que $S_N - K_N > X_N$ et $X_{N-1} > K_N$. Pour que ce vecteur X soit admissible on doit avoir $K_N^g + \sum_{k=i}^{N-1} (S_k^g - X_k) > 0$ pour $i < N$.

Donc $K_N^g + \sum_{k=i}^{N-2} (S_k^g - X_k) + S_{N-1}^g > X_{N-1}$ ce qui implique la nécessité d'avoir $\sum_{k=i}^{N-2} (S_k^g) + S_{N-1}^g > 0$ pour $i < N$ si l'on souhaite que l'argument K_N^g soit utile. En fait, avoir $S_{N-1}^g > 0$ suffit. Le vecteur $X = (-K_N^g - 1) \cdot 1_N + (K_N^g + 1) \cdot 1_{N-1}$ devient alors admissible et montre que l'argument K_N^g est utile.

- Pour que l'argument K_i^g (avec $1 < i < N$) soit strictement le plus petit de la fonction \min on doit avoir les contraintes $K_j^g + \sum_{k=i}^{j-1} (S_k^g - X_k) > K_i^g$ pour $j > i$ donc

$K_j^g + \sum_{k=i}^{j-1} S_k^g > K_i^g$. On doit aussi avoir $X_{i-1} > K_i^g$, or pour tout X admissible on a $K_i^g + S_{i-1}^g - X_{i-1} \geq 0$ donc $S_{i-1}^g > 0$. Sous ces conditions le vecteur $X = (-K_i^g - 1) \cdot 1_N + (K_i^g + 1) \cdot 1_{i-1}$ est admissible et montre que l'argument K_i^g est utile.

- Similairement pour que l'argument K_1^g soit utile on doit avoir $K_j^g + \sum_{k=1}^{j-1} S_k^g > K_1^g$ pour $1 < j \leq N$. On prend le vecteur $X = (-K_1^g - 1) \cdot 1_N$ qui est alors admissible, pour montrer que K_1^g est utile.

- Pour que l'argument $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ avec $i < j \leq N$ soit utile, il faut au moins

vérifier $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k) < K_l^g \otimes \bigotimes_{k=i}^{l-1} (S_k^g / X_k)$ pour $l > j$. On trouve alors les contraintes $K_l^g + \sum_{k=j}^{l-1} (S_k^g - X_k) > K_j^g$ équivalentes à la contrainte

$K_l^g + \sum_{k=j}^{l-1} S_k^g > K_j^g$. Il faut aussi vérifier $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k) < X_{i-1}$ et

$K_j^g \otimes \bigotimes_{k=i-1}^{j-1} (S_k^g / X_k) \geq 0$ pour $i > 1$, on trouve ainsi la contrainte $S_{i-1}^g > 0$. Si

toutes ces contraintes sont vérifiées alors l'état global

$X = \sum_{k=1}^{j-1} (S_k^g \cdot 1_k) + K_j^g \cdot 1_{j-1} - 1_N$ est admissible et il montre l'utilité de l'argument étudié.

- Pour montrer que l'argument $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile, on prend $X = S^g$ (le vecteur S^g). Si $S_{i-1}^g = 0$ l'argument précédent est inutile, ceci se démontre exactement comme cela a été fait pour le kanban étendu.
- Pour l'argument X_{i-1} , on prend $X = -1_N$.

□

Remarque :

Dans $F_i^g(X)$, tous les arguments de la fonction *min* utilisant K_i^g ne sont utiles que si les contraintes $K_j^g + \sum_{k=i}^{j-1} S_k^g > K_i^g$ pour $j > i$ sont vérifiées. S'il est souhaité que chaque paramètre K_i^g pour i allant de 1 à N puisse influencer la valeur d'au moins une fonction $F_i^g(X)$ il suffit que chaque contrainte $K_{i+1}^g + S_i^g > K_i^g$ soit respectée (par sommation de celles-ci on peut retrouver les contraintes $K_j^g + \sum_{k=i}^{j-1} S_k^g > K_i^g$). Ces contraintes sur le paramétrage sont connues (voir [BUZ 1989.a]).

□

Proposition 5.1 :

On considère deux lignes de N étages dont les processus de fabrication sont identiques et composés d'un stock tampon suivi d'un mécanisme quelconque de production d'où les pièces sortent une à une. La première ligne est gérée par une politique kanban généralisé et la seconde est gérée une politique kanban étendu. On a alors pour des paramétrages finis des politiques :

- La première ligne donnée a un comportement identique à la seconde pour un critère de type (a) si, et seulement si, le paramétrage du kanban étendu vérifie $K_i^e = S_i^e$ pour i allant de 1 à $N-1$. Le paramétrage de la politique kanban généralisé est alors défini par :

$$K_i^g = K_i^e \text{ et } S_i^g = S_i^e \text{ pour } i \text{ allant de } 1 \text{ à } N$$

- La seconde ligne donnée a un comportement identique à la première pour un critère de type (a) si et seulement si le paramétrage du kanban généralisé vérifie $K_N^g \geq S_N^g$ et $K_i^g = S_i^g$ pour i allant de 1 à $N-1$. Le paramétrage de la politique kanban étendu est alors défini par :

$$K_i^e = K_i^g \text{ et } S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N$$

□

Preuve :

Pour avoir des comportements identiques pour un critère de type (a) ces deux lignes doivent avoir des fonctions $F_i(X)$ identiques. Tous les arguments des fonctions min pouvant être utiles ont été répertoriés pour le kanban étendu et le kanban généralisé par les lemmes 5.1 et 5.2. Pour montrer l'égalité des fonctions $F_i^e(X)$ et $F_i^g(X)$ il faut observer une égalité de ces arguments utiles. Ainsi on trouve les conditions nécessaires et suffisantes suivantes à l'identité des politiques kanban étendu et kanban généralisé :

- L'argument K_i^e est toujours utile à la fonction min associée à $F_i^e(X)$ donc l'argument correspondant K_i^g devra lui aussi être utile dans $F_i^g(X)$. On a donc

$$K_{i+1}^g + S_i^g > K_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N-1,$$

$$S_{i-1}^g > 0 \text{ pour } i \text{ allant de } 2 \text{ à } N \text{ et}$$

$$K_i^e = K_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

- Pour $i < j \leq N$ l'argument $K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k)$ est toujours utile à la fonction min associée à $F_i^e(X)$. Toutes les conditions nécessaires à ce que l'argument

$K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ soit utile sont requises. On doit enfin vérifier

$$K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k) = K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k). \text{ On trouve alors}$$

$$K_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N-1.$$

- Pour $2 \leq i \leq N$ on a $S_{i-1}^g > 0$ donc pour $1 \leq i \leq N$ on doit avoir

$$\bigotimes_{k=i}^N (S_k^e / X_k) = \bigotimes_{k=i}^N (S_k^g / X_k), \text{ on en déduit alors}$$

$$S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

Il ne faut pas oublier que pour le kanban étendu on a $K_i^e \geq S_i^e$, et donc on a aussi $K_i^g \geq S_i^g$. Ce que nous venons d'obtenir permet de prouver la proposition.

□

Au niveau des $N-1$ premières mailles de ces deux lignes, les contraintes $K_i^g = S_i^g$ et $K_i^e = S_i^e$ signifient qu'il y a un contrôle identique à une politique kanban. Ainsi la proposition montre que dans le cas considéré, pour toute ligne dont la politique est de type kanban, sauf sur la dernière maille de type kanban étendu, il est possible de trouver une ligne au comportement identique dont la politique est de type kanban sauf sur la dernière maille de type kanban généralisé. La figure 5.2 illustre par un réseau de file d'attente cette inclusion de ce sous-ensemble de la politique kanban étendu dans ce sous-ensemble de la politique kanban généralisé.

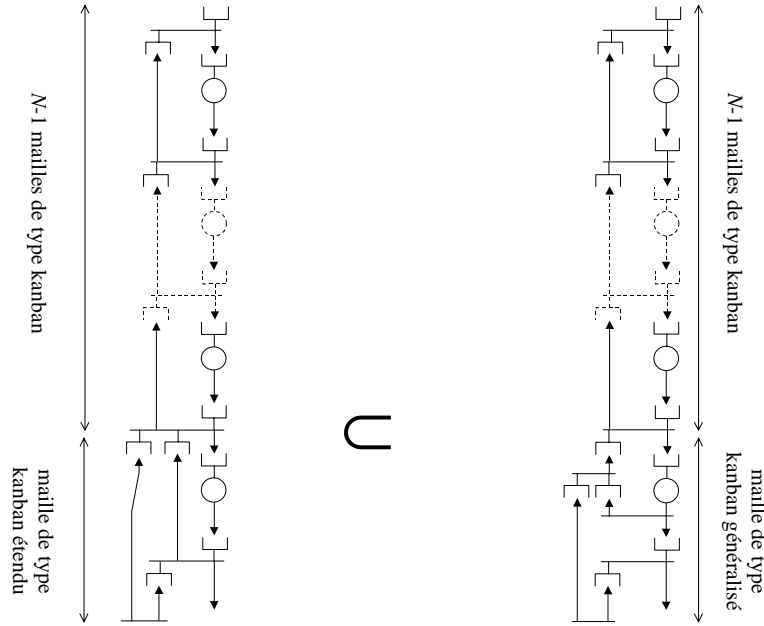


Figure 5.2 : Inclusion d'un sous-ensemble de la politique kanban étendu dans un sous-ensemble de la politique kanban généralisé (avec stock tampon, sortie une à une des pièces et paramétrage borné).

5.3.1.2 Processus de fabrication avec stock tampon et station monoserveur.

Les lignes considérées ici comportent N mailles, les processus de fabrication sont composés d'un stock tampon suivi d'une station monoserveur. Dans ce cas, pour un critère de type (a), il y a un même comportement entre une politique kanban étendu et une politique kanban généralisé, si et seulement si les valeurs $F_i^g(X)$ et $F_i^e(X)$ sont identiques pour i allant de 1 à N et cela pour chaque vecteur X admissible. Les résultats sont alors exactement les mêmes que dans la section précédente. Pour un critère de type (b), les deux politiques ont le même comportement si et seulement si les valeurs $M_i^g(X)$ et $M_i^e(X)$ sont identiques pour i allant de 1 à N et pour chaque vecteur X admissible. On a donc :

$$M_1^e(X) = 1 \oplus \bigoplus_{k=2}^N \left(\bigotimes_{j=1}^{k-1} (K_j^e / X_j) \otimes K_k^e \right) \oplus \bigotimes_{k=1}^N (S_k^e / X_k) \oplus K_1^e$$

égal à

$$M_1^g(X) = 1 \oplus \bigoplus_{j=2}^N \left(K_j^g \otimes \bigotimes_{k=1}^{j-1} (S_k^g / X_k) \right) \oplus \bigotimes_{k=1}^N (S_k^g / X_k) \oplus K_1^g$$

et pour $i > 1$ et $i \leq N$ on a

$$M_i^e(X) = 1 \oplus \bigoplus_{k=i+1}^N \left(\bigotimes_{j=i}^{k-1} (K_j^e / X_j) \otimes K_k^e \right) \oplus \bigotimes_{k=i}^N (S_k^e / X_k) \oplus K_i^e \oplus X_{i-1}$$

égal à

$$M_i^g(X) = 1 \oplus \bigoplus_{j=i+1}^N \left(K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k) \right) \oplus \bigotimes_{k=i}^N (S_k^g / X_k) \oplus K_i^g \oplus X_{i-1}$$

Chacune de ces formules est une fonction *min* dont les arguments sont des fonctions linéaires de X . On commence comme dans la section précédente par vérifier s'il n'y a pas des arguments inutiles. Il faut pour cela montrer qu'il existe un vecteur global admissible tel que l'argument considéré est nul alors que tous les autres sont non nuls.

Examinons pour $M_i^e(X)$ chaque argument de la fonction *min* :

- L'argument K_i^e est inutile car supérieur ou égal à l'argument 1.
- Pour l'argument X_{i-1} , on montre son utilité avec $X = -1_N$.
- Pour l'argument $\left(\bigotimes_{j=i}^{k-1} (K_j^e / X_j) \otimes K_k^e \right)$, on montre son utilité en employant

$$X = \sum_{\substack{j=1 \\ j \notin \{k-1, k\}}}^{N-1} (K_j^e \cdot 1_j) + (K_{k-1}^e + K_k^e) \cdot 1_{k-1} + \sum_{j=1}^N (S_j^e - K_j^e) \cdot 1_N.$$

- Pour l'argument $\bigotimes_{k=i}^N (S_k^e / X_k)$ on emploie $X = S^e$; si $S_{i-1}^e = 0$ l'argument est inutile (voir section précédente).

Examinons à présent pour $M_i^g(X)$ les arguments de sa fonction *min* :

- L'argument K_i^g est inutile car supérieur ou égal à l'argument 1.
- Pour que l'argument $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ avec $i < j \leq N$ soit utile, il faut au moins

vérifier les contraintes $K_l^g + \sum_{k=j}^{l-1} S_k^g > K_j^g$ pour $2 \leq j < l \leq N$ et la contrainte

$S_{i-1}^g > 0$ pour $2 \leq i \leq N-1$. Si toutes ces contraintes sont vérifiées alors l'état

global $X = \sum_{k=1}^{j-1} (S_k^g \cdot 1_k) + K_j^g \cdot 1_{j-1} - 1_N$ est admissible et il montre l'utilité de cet

argument.

- Pour montrer que l'argument $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile, on prend $X = S^g$. Si $S_{i-1}^g = 0$

l'argument précédent est inutile (voir section précédente).

- Pour l'argument X_{i-1} , on prend $X = -1_N$.

Proposition 5.2 :

On considère deux lignes de N étages dont les processus de fabrication sont identiques et composés d'un stock tampon suivi d'un monoserveur. La première ligne est gérée par une politique kanban généralisé et la seconde est gérée par une politique kanban étendu. On a alors pour des paramétrages finis des politiques :

- La première ligne donnée a un comportement identique à la seconde pour un critère de type (b) si et seulement si le paramétrage du kanban étendu vérifie $K_i^e = S_i^e$ pour i allant de 1 à $N-2$. Le paramétrage de la politique kanban généralisé est alors défini par :

$$K_N^g = K_N^e + K_{N-1}^e - S_{N-1}^e,$$

$$K_1^g \text{ positif non nul quelconque (on peut prendre } K_1^g = S_1^g),$$

$$K_i^g = K_i^e \text{ pour } i \text{ allant de } 2 \text{ à } N-1 \text{ et}$$

$$S_i^g = S_i^e \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

- La seconde ligne donnée a un comportement identique à la première pour un critère de type (b) si et seulement si le paramétrage du kanban généralisé vérifie $S_{N-1}^g \leq K_{N-1}^g < S_{N-1}^g + K_N^g$ et $K_i^g = S_i^g$ pour i allant de 2 à $N-2$. Le paramétrage de la politique kanban étendu est alors défini par :

$$K_N^e = K_N^g + S_{N-1}^g - K_{N-1}^e,$$

$$K_1^e = S_1^g$$

$$K_i^e = K_i^g \text{ pour } i \text{ allant de } 2 \text{ à } N-1 \text{ et}$$

$$S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

□

Preuve :

Tous les arguments des fonctions *min* pouvant être utiles sont répertoriés avant la proposition. Pour montrer l'égalité des fonctions $M_i^e(X)$ et $M_i^g(X)$ il faut observer une identité de ces arguments. Ainsi on trouve les conditions nécessaires et suffisantes suivantes à l'identité des politiques kanban étendu et kanban généralisé :

- Pour $1 \leq i < j \leq N$ l'argument $K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k)$ est toujours utile à la fonction *min* associée à $M_i^e(X)$. Toutes les conditions nécessaires à ce que l'argument

$K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ soit utile sont donc requises. On doit enfin vérifier

$$K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k) = K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k). \text{ On trouve alors}$$

$$S_i^g > 0 \text{ pour } i \text{ allant de } 1 \text{ à } N-2,$$

$$K_{i+1}^g + S_i^g > K_i^g \text{ pour } i \text{ allant de } 2 \text{ à } N-1,$$

$$K_N^e + K_{N-1}^e = K_N^g + S_{N-1}^g,$$

$$K_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N-2 \text{ et}$$

$$K_i^e = K_i^g \text{ pour } i \text{ allant de } 2 \text{ à } N-1.$$

- Pour $2 \leq i \leq N-1$ on a $S_{i-1}^g > 0$ donc pour $1 \leq i \leq N-1$ on doit avoir

$$\bigotimes_{k=i}^N (S_k^e / X_k) = \bigotimes_{k=i}^N (S_k^g / X_k), \text{ on en déduit alors}$$

$$S_N^e + S_{N-1}^e = S_N^g + S_{N-1}^g \text{ et } S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N-2.$$

Si $S_{N-1}^e = 0$ ou $S_{N-1}^g = 0$ alors $S_{N-1}^e = S_{N-1}^g = 0$ et $S_N^e = S_N^g$

sinon $S_{N-1}^g > 0$ et $S_{N-1}^e > 0$ ce qui implique $S_N^e - X_N = S_N^g - X_N$.

On obtient donc :

$$S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

Il ne faut pas oublier que pour le kanban étendu $K_i^e \geq S_i^e$ donc on a aussi $K_i^g \geq S_i^g$.
Ce que nous venons d'obtenir démontre la proposition. \square

Au niveau des $N-2$ premières mailles de la première ligne, la contrainte $K_i^e = S_i^e$ signifie qu'il y a un contrôle identique à une politique kanban. Ainsi le premier point de la proposition montre que dans le cas considéré, toute ligne dont la politique est de type kanban sauf sur les deux dernières mailles où l'on a une politique de type kanban étendu, il est possible de trouver une ligne au comportement identique dont la politique est de type kanban sauf sur les deux dernières mailles de type kanban généralisé. Le second point de la proposition montre que l'inverse n'est pas possible en général puisque les contraintes $S_{N-1}^g \leq K_{N-1}^g$ et $K_{N-1}^g < S_{N-1}^g + K_N^g$ doivent être vérifiées. La figure 5.3 illustre par un réseau de files d'attente cette inclusion de ce sous-ensemble de la politique kanban étendu dans ce sous-ensemble de la politique kanban généralisé.

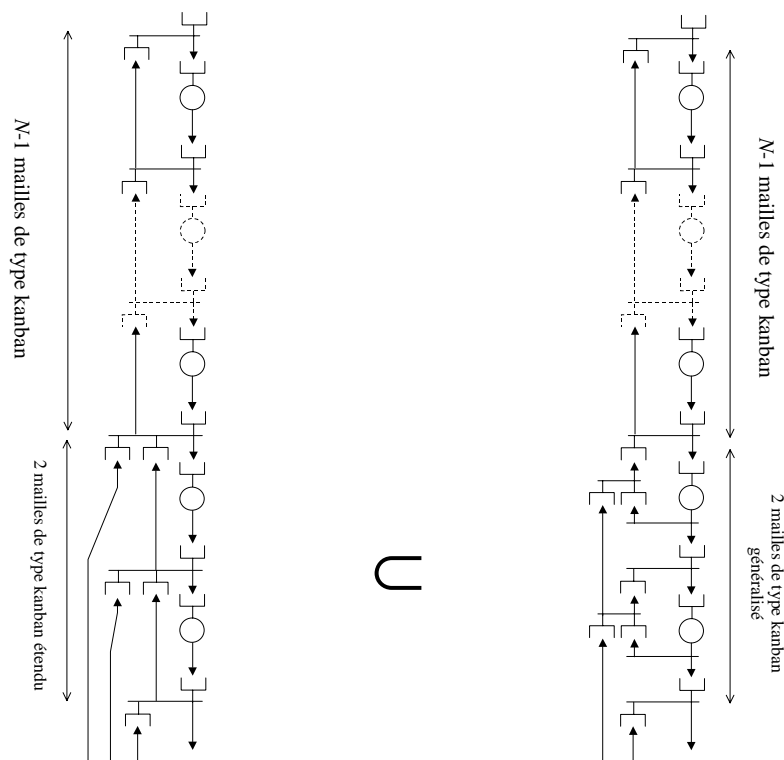


Figure 5.3 : Inclusion d'un sous-ensemble de la politique kanban étendu dans un sous-ensemble de la politique kanban généralisé (avec stock tampon, monoserveur et paramétrage borné).

5.3.2 Comparaisons avec des paramètres K_i de valeur infinie.

Nous avons rappelé précédemment qu'un système kanban étendu ayant un nombre illimité de kanbans est équivalent à un système de type basestock. Plusieurs équivalences similaires à celles que nous venons de voir sont possibles entre le kanban étendu et le kanban généralisé lorsqu'un paramétrage à valeurs infinies est autorisé. Nous montrons dans cette section qu'il est possible d'en faire l'inventaire.

Pour montrer une équivalence entre politiques, nous comparons des formules composées d'une fonction \min . En donnant une valeur infinie à un paramètre, certains arguments de ces fonctions \min deviennent infinis et peuvent être supprimés. On peut rappeler que pour l'algèbre $(\min,+)$, l'infini est l'élément neutre de l'addition et l'élément absorbant de la multiplication.

Seuls les paramètres K_i seront portés à des valeurs infinies car lorsqu'un paramètre S_i est infini on observe que le système se met à produire sans fin sur une portion amont. Lorsqu'un paramètre S_i est infini, on observe pour les politiques kanban étendu et kanban généralisé, que la marche ou l'arrêt des processus de fabrication F_j pour $j \leq i$ ne dépend pas de X_k pour $k \geq i$. Pour s'en convaincre il suffit de supprimer tous les arguments infinis des fonctions \min . La politique consiste alors à produire sans interruption dans la portion de ligne constituée des mailles 1 à i . Le système se comporte donc comme s'il y avait un déficit sans borne de produit dans la maille i . Le cas S_i infini est donc considéré sans intérêt car le système ne produit pas en fonction de la demande.

5.3.2.1 Processus de fabrication avec stock tampon et pièces sortant une à une.

On reprend ici le cas étudié en section 5.3.1.1 mais en considérant cette fois-ci que les paramétrages peuvent être infinis. Dans la proposition qui suit, l'indice de certaines composantes peut devenir inférieur à 1 ou supérieur à N , ces composantes n'ont alors pas de sens et les formules dans lesquelles elles se situent doivent être ignorées, cela permet de prendre en compte certains cas particuliers.

Proposition 5.3 :

On considère deux lignes de N étages dont les processus de fabrication sont identiques et composés d'un stock tampon suivi d'un mécanisme de production quelconque d'où les pièces sortent une à une. La première ligne est gérée par une politique kanban généralisé et la seconde est gérée par une politique kanban étendu. Alors pour un paramétrage de ces politiques où au moins un des paramètres relatif au nombre de kanbans est infini, on a :

- La première ligne donnée a un comportement identique à la seconde pour un critère de type (a) si, et seulement si, le paramétrage du kanban étendu vérifie $K_i^e = \infty$ pour i

allant de $j+1$ à N et $K_i^c = S_i^c$ pour i allant de 1 à $j-1$, avec j compris entre 1 et N . Le paramétrage de la politique kanban généralisé est alors défini par :

$$K_i^g = K_i^c \text{ et } S_i^g = S_i^c \text{ pour } i \text{ allant de } 1 \text{ à } N$$

- La seconde ligne donnée a un comportement identique à la première pour un critère de type (a) si, et seulement si, pour j compris entre 1 et N , le paramétrage du kanban généralisé vérifie $K_i^g = \infty$ pour i allant de $j+1$ à N , $K_j^g \geq S_j^g$ et $K_i^g = S_i^g$ pour i allant de 1 à $j-1$. Le paramétrage de la politique kanban étendu est alors défini par :

$$K_i^c = K_i^g \text{ et } S_i^c = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N$$

□

Preuve :

Comme pour le cas où le paramétrage est fini il faut montrer que pour chaque vecteur X admissible les valeurs $F_i^g(X)$ et $F_i^c(X)$ sont identiques pour i allant de 1 à N . Avant de regarder si pour ces formules les arguments des fonctions *min* sont identiques on va commencer par vérifier s'il n'y en a pas qui sont inutiles.

Examinons pour $F_i^c(X)$ chaque argument de la fonction *min* comme cela est fait dans la démonstration du lemme 5.1 :

- L'argument K_i^c est inutile si et seulement si il est infini.
- Avec l'argument X_{i-1} , on prend $X = -1_N$ pour montrer qu'il est utile.
- L'argument $\left(\bigotimes_{j=i}^{k-1} (K_j^c / X_j) \otimes K_k^c \right)$ est utile si $K_j^c \neq \infty$ pour $i \leq j \leq k$.
- L'argument $\bigotimes_{k=i}^N (S_k^c / X_k)$ est utile si et seulement si $S_{i-1}^c > 0$.

Examinons à présent pour $F_i^g(X)$ les arguments de sa fonction *min* :

- L'argument K_1^g est utile s'il n'est pas infini et si $K_j^g + \sum_{k=1}^{j-1} S_k^g > K_1^g$.
- L'argument K_i^g est utile si :

$$K_j^g + \sum_{k=i}^{j-1} S_k^g > K_i^g \text{ pour } j > i, S_{i-1}^g > 0 \text{ et } K_i^g < \infty.$$

- Pour que l'argument $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ avec $i < j \leq N$ soit utile, il faut vérifier :

$$K_l^g + \sum_{k=j}^{l-1} S_k^g > K_j^g \text{ avec } j < l \leq N, S_{i-1}^g > 0 \text{ si } i > 1 \text{ et } K_j^g \neq \infty$$

- L'argument $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile si et seulement si $S_{i-1}^g > 0$.
- L'argument X_{i-1} reste toujours utile.

Pour montrer l'égalité des fonctions $F_i^e(X)$ et $F_i^g(X)$ il faut observer une égalité de ces arguments. Ainsi on trouve les conditions nécessaires et suffisantes suivantes à l'identité des politiques kanban étendu et kanban généralisé :

- Si $K_i^e \neq \infty$ alors l'argument K_i^e en $F_i^e(X)$ est utile, ceci implique que l'argument K_i^g doit lui aussi être utile en $F_i^g(X)$. On a donc

$$K_{i+1}^g + S_i^g > K_i^g \text{ si } K_i^e \neq \infty, \text{ avec } i \text{ allant de } 1 \text{ à } N-1,$$

$$S_{i-1}^g > 0 \text{ si } K_i^e \neq \infty, \text{ avec } i \text{ allant de } 2 \text{ à } N \text{ et}$$

$$K_i^e = K_i^g \text{ si } K_i^e \neq \infty, \text{ avec } i \text{ allant de } 1 \text{ à } N.$$

- Si $K_i^g = \infty$ alors l'argument K_i^g est inutile et donc l'argument K_i^e doit lui aussi être inutile. On a donc

$$K_i^g = K_i^e \text{ si } K_i^g = \infty, \text{ avec } i \text{ allant de } 1 \text{ à } N.$$

- Si $K_k^e = \infty$ alors pour chaque i, j vérifiant $i < j$ et $i \leq k \leq j$, l'argument

$$K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k) \text{ est inutile. Il ne faut pas alors que l'argument}$$

$$K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k) \text{ soit utile. Pour cela on choisit de poser soit } S_{i-1}^g = 0, K_j^g = \infty$$

ou $K_l^g + \sum_{k=j}^{l-1} S_k^g \leq K_j^g$ avec $j < l \leq N$. On peut toujours avoir $i = 1$ et $j = N$, on

est alors obligé de poser $K_N^g = \infty$ car les autres choix possibles ne sont pas définis. On a $K_N^g = \infty$. Pour $k < N$ on peut toujours avoir $i = 1$ et $j = N - 1$, on est alors obligé de poser $K_{N-1}^g = \infty$ car pour poser $K_N^g + S_{N-1}^g \leq K_{N-1}^g$ il faut avoir $K_{N-1}^g = \infty$. Par une démonstration par récurrence on montre que si $K_k^e = \infty$ alors $K_l^g = \infty$ pour $k \leq l \leq N$. On a donc :

$$K_l^g = K_l^e = \infty \text{ pour } k \leq l \leq N \text{ si } K_k^e = \infty.$$

- Si $K_j^e \neq \infty$ alors $K_k^e \neq \infty$ pour $k > j$, donc l'argument $K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k)$ contenu en $F_i^e(X)$ est utile. Ainsi on a $K_j^e \otimes \bigotimes_{k=i}^{j-1} (K_k^e / X_k) = K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$.

On trouve après simplification $K_i^e = S_i^g$ pour $K_{i+1}^e \neq \infty$ et $i > N$.

- $\bigotimes_{k=i}^N (S_k^g / X_k)$ et $\bigotimes_{k=i}^N (S_k^e / X_k)$ sont utiles si et seulement si $i = 1$ ou si respectivement $S_{i-1}^g > 0$ et $S_{i-1}^e > 0$. Pour $i \geq 2$, $K_i^e \neq \infty$ implique $S_{i-1}^g > 0$. Ainsi, pour $i = 1$

ou pour i tel que $K_i^e \neq \infty$, l'argument $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile, on doit alors avoir

$$\bigotimes_{k=i}^N (S_k^e / X_k) = \bigotimes_{k=i}^N (S_k^g / X_k). \text{ On trouve donc après simplification } S_i^e = S_i^g \text{ pour}$$

$$K_{i+1}^e \neq \infty \text{ et } i > N.$$

- Pour $i \geq 2$ tel que $K_i^g = K_i^e = \infty$, il est possible d'avoir $S_{i-1}^g = 0$ ou $S_{i-1}^e = 0$.

Ceci rend alors respectivement $\bigotimes_{k=i}^N (S_k^g / X_k)$ ou $\bigotimes_{k=i}^N (S_k^e / X_k)$ inutile. Si l'un de ces

deux arguments est inutile l'autre le devient à son tour. Or ces deux arguments ne peuvent être inutiles que si $S_{i-1}^e = 0$ et $S_{i-1}^g = 0$. Donc $S_{i-1}^g = 0$ si et seulement si $S_{i-1}^e = 0$.

- Supposons $S_{i-1}^g = S_{i-1}^e = 0$ alors $\bigotimes_{k=i}^N (S_k^g / X_k)$ et $\bigotimes_{k=i}^N (S_k^e / X_k)$ sont inutiles. Soit j

le plus grand entier strictement inférieur à i , tel que $S_{j-1}^e > 0$ ou tel que $j=1$, on a

alors l'argument $\bigotimes_{k=j}^N (S_k^e / X_k)$ qui est utile et par suite on en déduit $\bigotimes_{k=j}^N S_k^e = \bigotimes_{k=j}^N S_k^g$.

Après suppression des paramètres nuls on trouve $\bigotimes_{k=i}^N S_k^e = \bigotimes_{k=i}^N S_k^g$. On obtient alors :

$$S_i^e = S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N.$$

Le premier point est démontré. En utilisant $K_i^e \geq S_i^e$ on démontre le deuxième point.

□

Au niveau des $j - 1$ premières mailles de ces deux lignes, les contraintes $K_i^g = S_i^g$ et $K_i^e = S_i^e$ signifient qu'il y a un contrôle identique à une politique kanban. De même, dans les $N - j$ dernières mailles les contraintes $K_i^g = \infty$ et $K_i^e = \infty$ signifient qu'il y a un contrôle identique à une politique basestock.

Ainsi la proposition montre que dans le cas considéré, seules les lignes dont la politique est de type kanban étendu dans une maille avec en amont et en aval de celle-ci des mailles dont les politiques sont respectivement de type kanban et basestock, permettent de trouver une ligne au comportement identique dont la politique est de type kanban généralisé. Le second point de la proposition montre que l'inverse n'est pas possible en général puisque la contrainte $K_j^g \geq S_j^g$ doit être vérifiée. La figure 5.4 illustre par un réseau de files d'attente cette inclusion de ce sous-ensemble de la politique kanban étendu dans ce sous-ensemble de la politique kanban généralisé.

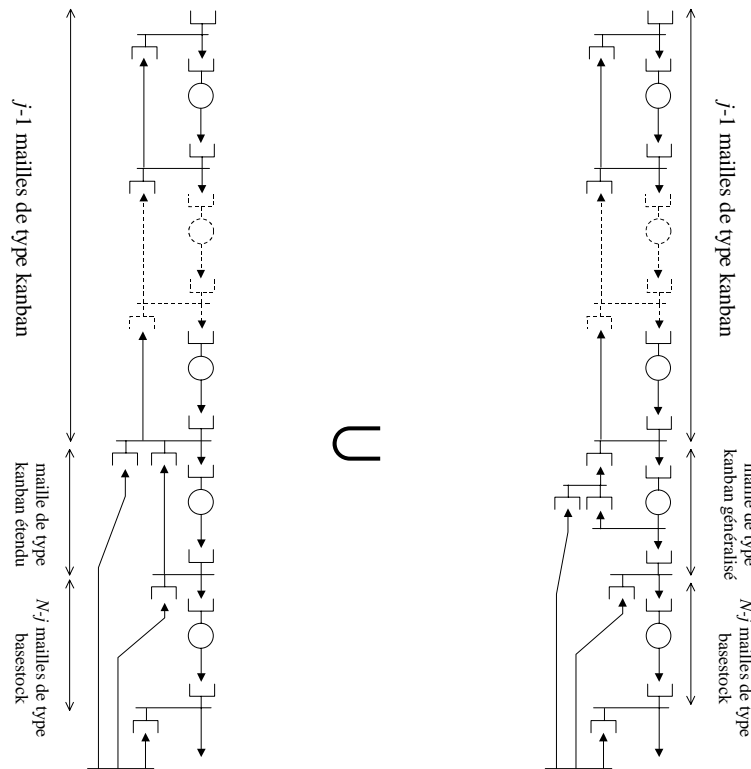


Figure 5.4 : Inclusion d'un sous-ensemble de la politique kanban étendu dans un sous-ensemble de la politique kanban généralisé (avec stock tampon, sortie une à une des pièces et paramétrage non borné).

5.3.2.2 Processus de fabrication avec stock tampon et station monoserveur.

On reprend ici le cas étudié en section 5.3.1.2 mais en considérant cette fois-ci que les paramétrages peuvent être infinis. Dans la proposition qui suit, l'indice de certaines composantes peut devenir strictement inférieur à 1 ou strictement supérieur à N . Ces composantes n'ont alors pas de sens et les formules dans lesquelles elles se situent doivent être ignorées. Cela permet de prendre en compte certains cas où l'on considère la première ou la dernière maille de la ligne de production.

Proposition 5.4 :

On considère deux lignes de N étages dont les processus de fabrication sont identiques et composés d'un stock tampon suivi d'un mécanisme de production quelconque d'où les pièces sortent une à une. La première ligne est gérée par une politique kanban généralisé et la seconde est gérée par une politique kanban étendu. Alors pour un paramétrage de ces politiques où au moins un des paramètres relatif au nombre de kanbans est infini on a :

- La première ligne donnée a un comportement identique à la seconde pour un critère de type (b) si et seulement si le paramétrage du kanban étendu vérifie $K_i^e = \infty$ pour i

allant de $j+1$ à N et $K_i^e = S_i^e$ pour i allant de 1 à $j-2$ avec j compris entre 1 et N . Le paramétrage de la politique kanban généralisé est alors défini par :

$$\begin{aligned} K_i^g &= K_i^e \text{ pour } i \text{ allant de } 1 \text{ à } j-1, \\ K_j^g &= K_j^e + K_{j-1}^e - S_{j-1}^e, \\ K_i^g &= K_i^e = \infty \text{ pour } i \text{ allant de } j+1 \text{ à } N, \text{ et} \\ S_i^g &= S_i^e \text{ pour } i \text{ allant de } 1 \text{ à } N. \end{aligned}$$

- La seconde ligne donnée a un comportement identique à la première pour un critère de type (b) si et seulement si pour j compris entre 1 et N , le paramétrage du kanban généralisé vérifie $K_i^g = \infty$ pour i allant de $j+1$ à N , $S_{j-1}^g \leq K_{j-1}^g < S_{j-1}^g + K_j^g$ et $K_i^g = S_i^g$ pour i allant de 2 à $j-2$. Le paramétrage de la politique kanban étendu est alors défini par :

$$\begin{aligned} K_1^e &= S_1^g, \\ K_i^e &= K_i^g \text{ pour } i \text{ allant de } 2 \text{ à } j-1, \\ K_j^e &= K_j^g + S_{j-1}^g - K_{j-1}^g, \\ K_i^e &= K_i^g = \infty \text{ pour } i \text{ allant de } j+1 \text{ à } N \text{ et} \\ S_i^e &= S_i^g \text{ pour } i \text{ allant de } 1 \text{ à } N. \end{aligned}$$

□

Preuve :

Comme pour le cas où le paramétrage est fini il faut montrer que, pour chaque vecteur X admissible, les valeurs $M_i^g(X)$ et $M_i^e(X)$ sont identiques pour i allant de 1 à N . Avant de regarder si pour ces formules les arguments des fonctions *min* sont identiques on va commencer par vérifier s'il n'y en a pas qui sont inutiles.

Examinons pour $M_i^e(X)$ chaque argument de la fonction *min* :

- L'argument K_i^e est inutile.
- L'argument $\left(\bigotimes_{j=i}^{k-1} (K_j^e / X_j) \otimes K_k^e \right)$ est utile si $K_j^e \neq \infty$ pour $i \leq j \leq k$.
- L'argument $\bigotimes_{k=i}^N (S_k^e / X_k)$ est utile si $S_{i-1}^e > 0$ ou si $i=1$

Examinons à présent pour $M_i^g(X)$ les arguments de sa fonction *min* :

- L'argument K_i^g est inutile.
- Pour que l'argument $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ avec $i < j \leq N$ soit utile, il faut vérifier :

$$K_l^g + \sum_{k=j}^{l-1} S_k^g > K_j^g \text{ avec } j < l \leq N, S_{i-1}^g > 0 \text{ si } i > 1 \text{ et } K_j^g \neq \infty$$

- L'argument $\bigotimes_{k=i}^N (S_k^g / X_k)$ est utile si $S_{i-1}^g > 0$ ou si $i=1$.

Pour montrer l'égalité des fonctions $M_i^c(X)$ et $M_i^g(X)$ il faut observer une égalité des arguments utiles. Ainsi on peut trouver les conditions nécessaires et suffisantes à l'identité des politiques kanban étendu et kanban généralisé :

- Pour $i < j \leq N$ l'argument $K_j^c \otimes \bigotimes_{k=i}^{j-1} (K_k^c / X_k)$ est inutile s'il existe $K_k^c = \infty$ avec $i \leq k \leq j$. Pour qu'il y ait égalité des fonctions *min* il faut, de façon symétrique, que l'argument $K_j^g \otimes \bigotimes_{k=i}^{j-1} (S_k^g / X_k)$ soit inutile. Pour cela on choisit soit de poser

$$S_{i-1}^g = 0, \quad K_j^g = \infty \quad \text{ou} \quad K_l^g + \sum_{k=j}^{l-1} S_k^g \leq K_j^g \quad \text{avec} \quad j < l \leq N.$$

Pour $i = 1$ et $j = N$ on est obligé de choisir $K_N^g = \infty$. Pour $i = 1$ et $j = N - 1$ on est obligé de choisir $K_{N-1}^g = \infty$ et on obtient alors $K_N^g + S_{N-1}^g \leq K_{N-1}^g$. Par une démonstration par récurrence on montre que si $K_k^c = \infty$ alors $K_l^g = \infty$ pour $k \leq l \leq N$. En supposant l tel que $K_{l+1}^c = \infty$ et $K_l^c \neq \infty$ on a donc

$$K_k^g = K_k^c = \infty \quad \text{pour} \quad l < k \leq N$$

$$S_i^g > 0 \quad \text{pour} \quad i \text{ allant de } 1 \text{ à } l-2,$$

$$K_{i+1}^g + S_i^g > K_i^g \quad \text{pour} \quad i \text{ allant de } 2 \text{ à } l-1,$$

$$K_l^c + K_{l-1}^c = K_l^g + S_{l-1}^g,$$

$$K_i^c = S_i^g \quad \text{pour} \quad i \text{ allant de } 1 \text{ à } l-2 \text{ et}$$

$$K_i^c = K_i^g \quad \text{pour} \quad i \text{ allant de } 2 \text{ à } l-1.$$

- Pour $1 \leq i \leq l - 2$ on a $S_i^g > 0$ donc pour $1 \leq i \leq l - 1$ on doit avoir

$$\bigotimes_{k=i}^N (S_k^c / X_k) = \bigotimes_{k=i}^N (S_k^g / X_k). \quad \text{Si } S_{i-1}^c = 0 \text{ alors } \bigotimes_{k=i}^N (S_k^c / X_k) \text{ est inutile, de même}$$

si $S_{i-1}^g = 0$ alors $\bigotimes_{k=i}^N (S_k^g / X_k)$ est inutile. Lorsqu'un de ces deux arguments est

inutile, par symétrie, l'autre doit l'être aussi, or il ne peuvent être tous deux inutiles que si $S_{i-1}^c = 0$ et $S_{i-1}^g = 0$. Donc lorsque $S_{i-1}^c = 0$ ou $S_{i-1}^g = 0$ on a

$$S_{i-1}^c = S_{i-1}^g = 0. \quad \text{En montrant qu'avec } S_{i-1}^c = 0 \text{ on a } \bigotimes_{k=i}^N (S_k^c / X_k) = \bigotimes_{k=i}^N (S_k^g / X_k)$$

on arrive à obtenir :

$$S_i^c = S_i^g \quad \text{pour} \quad i \text{ allant de } 1 \text{ à } N.$$

Le premier point est démontré. En utilisant $K_i^c \geq S_i^c$ on démontre le deuxième point.

□

Au niveau des $j - 2$ premières mailles de ces deux lignes, les contraintes $K_i^g = S_i^g$ et $K_i^c = S_i^c$ signifient qu'il y a un contrôle identique à une politique kanban. De même, dans les $N - j$ dernières mailles les contraintes $K_i^g = \infty$ et $K_i^c = \infty$ signifient qu'il y a un contrôle identique à une politique basestock. Ainsi la proposition montre que dans le cas considéré,

seules les lignes dont la politique est de type kanban étendu dans deux mailles avec en amont et en aval de celle-ci des mailles dont les politiques sont respectivement de type kanban et basestock, permettent de trouver une ligne au comportement identique dont la politique est de type kanban généralisé. Le second point de la proposition montre que l'inverse n'est pas possible en général car les contraintes $S_{j-1}^g \leq K_{j-1}^g$ et $K_{j-1}^g < S_{j-1}^g + K_j^g$ doivent être vérifiées. La figure 5.5 illustre par un réseau de files d'attente cette inclusion de ce sous-ensemble de la politique kanban étendu dans ce sous-ensemble de la politique kanban généralisé.

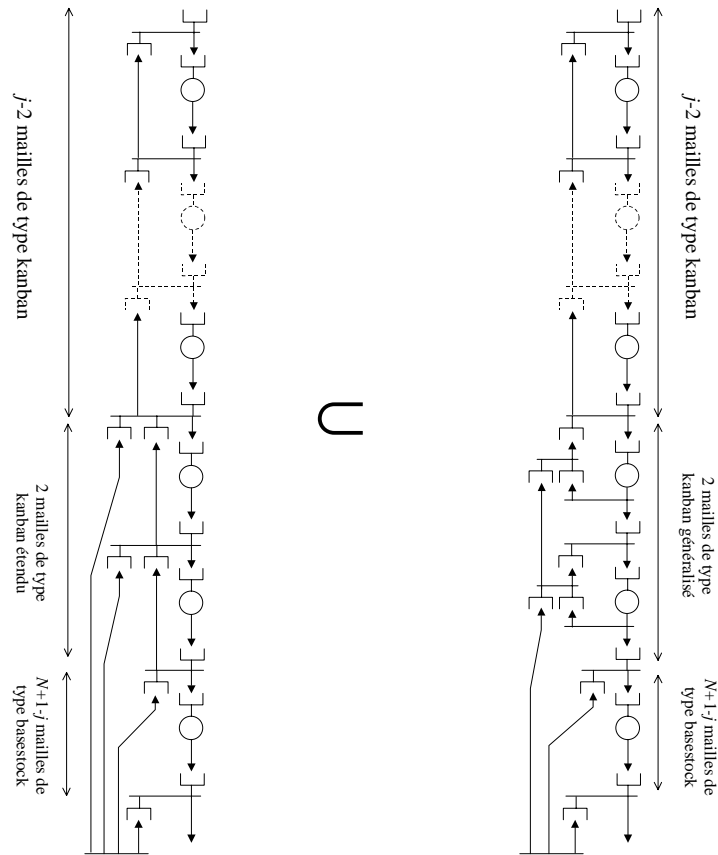


Figure 5.5 : Inclusion d'un sous-ensemble de la politique kanban étendu dans un sous-ensemble de la politique kanban généralisé (avec stock tampon, monoserveur et paramétrage non borné).

5.4 Conclusion sur la comparaison des politiques

En utilisant la valeur des F_i en fonction de X , nous avons pu comparer de façon plus approfondie les deux politiques kanban étendu et kanban généralisé. Pour différentes hypothèses, on a pu trouver le plus grand sous-ensemble de systèmes kanban étendu ayant un comportement identique à un système kanban généralisé. Ces systèmes kanban étendu sont en général composé de mailles de type kanban en amont suivi suivant le cas d'une à deux mailles

consécutives gérées par une politique kanban étendu. Cette structure générale varie suivant les cas étudiés :

Si l'on considère des systèmes contenant plusieurs mailles en série, dont les processus de fabrication sont composés d'un stock tampon et d'une station quelconque d'où les pièces sortent une à une, alors le système est composé de mailles kanban avec au plus deux mailles de type kanban étendu en fin de ligne.

Si les processus de fabrication sont composés d'un stock tampon suivi d'un monoserveur alors pour qu'une équivalence soit possible le système doit être composé de mailles kanban avec au plus une mailles de type kanban étendu en fin de ligne.

Si un paramétrage infini du nombre de kanban est autorisé alors la maille ou les deux mailles de type kanban étendu ne sont pas forcément situées en fin de ligne. Elle peuvent être en milieu de ligne. Ces mailles sont alors suivies jusqu'en fin de ligne de mailles de type basestock afin qu'une équivalence soit possible avec un système kanban généralisé. Chacun des cas est illustré dans les figures 5.2, 5.3, 5.4 et 5.5.

Nous avons obtenu de nouvelles équivalences de comportement entre les politiques kanban étendu et kanban généralisé, tout en retrouvant les équivalences de comportement déjà connues. La méthode que nous avons utilisée, a l'avantage de pouvoir exhiber toutes les équivalences de comportement possibles entre deux politiques exprimées par la formulation dite canonique. On peut donc conclure que les politiques kanban étendu et kanban généralisé induisent, la plupart du temps, des comportements différents dans les systèmes qu'elles gèrent.

Jusqu'à présent nous nous sommes contentés de comparer l'identité du comportement des systèmes face à tel ou tel critère. Nous allons maintenant nous intéresser aux coûts des systèmes pour une distribution donnée des événements.

Comparaison de coûts entre des politiques sous-optimales avec pour référence la politique optimale.

6.1 Introduction

Dans cette section nous comparons plusieurs politiques à flux tiré pour différents systèmes de production. Pour effectuer ces comparaisons, nous définissons un coût qui pourra être calculé pour chaque politique.

Les coûts considérés augmentent proportionnellement au nombre de pièces présentes dans les lignes de production et proportionnellement au nombre de demandes non servies à temps. Il s'agit ainsi de diminuer les coûts dus à la possession de pièces et ceux dus aux retards dans le service de la demande (perte de clientèle, ...).

La politique à flux tiré dont le coût est minimal sera appelée la politique optimale. Les politiques telles que le kanban, le basestock, le kanban étendu ou généralisé seront dites sous-optimales car elles sont plus coûteuses que la politique optimale. Cette politique optimale nous servira de référence lors de comparaisons entre les politiques sous-optimales.

Nous avons étudié plusieurs types de systèmes mais principalement des lignes composées de deux étages. Pour une ligne de deux étages, un vecteur d'état $X = (X_1, X_2)$ permet de comptabiliser le nombre de produits finis si X_2 est positif, le nombre de demandes non servies si X_2 est négatif, et X_1 comptabilise le nombre de produits intermédiaires compris entre les sorties des deux processus de fabrication. On s'est intéressé plus particulièrement au système étudié dans l'article de Veatch et Wein [VEA 1994] où chacun des deux étages est constitué d'une file d'attente ./M/1 (voir figure 6.1). La station de ces files d'attente a un taux de service μ_i contrôlable qui varie entre 0 et $\bar{\mu}_i$ en fonction du pilotage. Les demandes arrivent selon un processus de Poisson avec un taux λ .

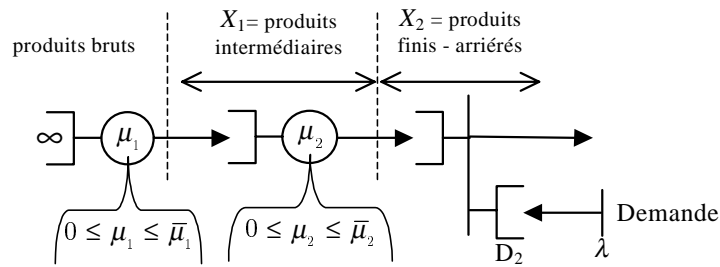


Figure 6.1 : Une ligne de deux mailles avec monoserveurs dont le taux de service est fonction du pilotage.

On s'intéresse à l'espace d'état du vecteur X et aux transitions qui font évoluer un vecteur d'état X lors de l'occurrence d'un événement. Au taux de service μ_1 , on associe la transition qui transforme X en $X + d_{\mu_1}$ avec $d_{\mu_1} = 1_1$ (1_i étant le vecteur dont la composante i vaut 1 alors que les autres sont nulles) ; ceci correspond à la production d'un produit intermédiaire. Au taux μ_2 est associée la transition qui transforme X en $X + d_{\mu_2}$ avec $d_{\mu_2} = 1_2 - 1_1$; ceci correspond à la production d'un produit fini à partir d'un produit intermédiaire. Au taux λ est associée la transition qui transforme X en $X + d_\lambda$ avec $d_\lambda = -1_2$; ceci correspond à l'arrivée d'une demande qui retire un produit fini, ou qui augmente l'arriéré s'il n'y a plus de produit fini. La figure 6.2 illustre chacune de ces transitions sur l'espace d'état du vecteur X .

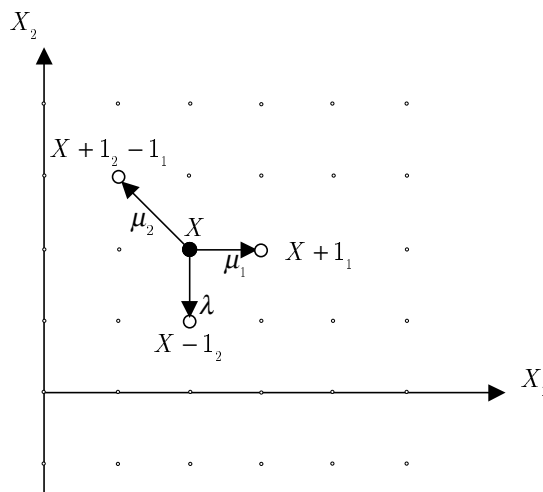


Figure 6.2 : Transitions possibles pour une ligne de deux étages.

Nous verrons, dans la section 6.2, que la commande optimale effectuée sur les taux de service est du type « bang-bang ». En d'autres termes, soit la station i marche au maximum avec $\mu_i = \bar{\mu}_i$, soit elle est à l'arrêt avec $\mu_i = 0$. Le système de la figure 6.1 est piloté par une commande directe des taux de service, cependant si on se limite à des politiques de pilotage de type « bang-bang » le système illustré sur la figure 6.3 peut être considéré comme équivalent. Ce système de deux étages se compose successivement d'un stock infini de produits bruts, d'un processus de fabrication F_1 , d'un stock de produits intermédiaires P_1 ,

d'un processus de fabrication F_2 et d'un stock de produits finis P_2 . Les processus de fabrication sont composés d'un stock tampon suivi d'un monoserveur. La commande $\mu_i = \bar{\mu}_i$ s'obtient en libérant une pièce en F_i et la commande $\mu_i = 0$ s'obtient en bloquant le franchissement de la station de synchronisation qui précède F_i .

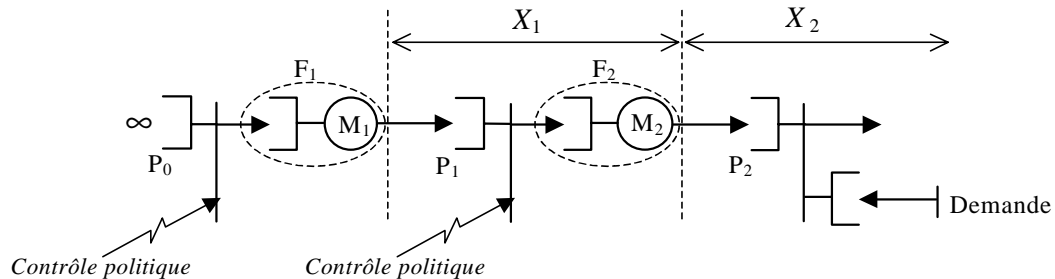


Figure 6.3 : Ligne de deux mailles où le contrôle de la production s'effectue au niveau de la libération des pièces dans chacun des processus de fabrication.

D'autres systèmes ont aussi été étudiés. On s'est intéressé au cas de stations produisant par lots. Nous avons aussi étudié les effets d'un délai de livraison autorisant, sans coûts supplémentaires, la non satisfaction immédiate de la demande. Nous avons distingué deux cas. Nous nous sommes d'abord intéressés à des systèmes où il est possible de servir la demande avant la fin de ce délai de livraison (figure 6.11) puis à des systèmes où il faut attendre la fin du délai de livraison pour servir une demande (figure 6.17). Enfin un système comprenant deux mailles suivies d'une maille d'assemblage a aussi été considéré (voir figure 6.30).

Les méthodes de calcul des coûts utilisées sont toutes basées sur la programmation dynamique, qui est incontournable pour la détermination d'une politique optimale. Deux catégories de coûts, inhérentes à cette méthode, sont étudiées. La programmation dynamique peut permettre de caractériser l'aspect général des politiques optimales. Il sera donc nécessaire d'expliquer cette méthode de calcul (section 6.2) avant de commencer l'étude de différents systèmes de production et de les comparer dans les sections suivantes.

6.2 Calcul de politiques optimales par programmation dynamique.

Les méthodes de programmation dynamique utilisent le principe d'optimalité [BEL 1957], [WEB 2001] afin de trouver quelle sera la commande optimale qui minimisera les coûts du système considéré. Ce principe affirme que pour tout point d'une trajectoire optimale, la trajectoire qui suit ce point est optimale si l'on considère des trajectoire partant de ce point.

De ce principe, on obtient une équation de programmation dynamique, ou de Bellman, qui permet de calculer le coût de la trajectoire optimale partant d'un point x à un instant t en

fonction du coût des trajectoires optimales partant des différents points atteignables à l'instant $t+1$.

Notons $V(x, t)$ le coût de la trajectoire optimale passant par x à l'instant t . Les coûts, en un point x lorsqu'un contrôle u est exercé entre les instants t et $t+1$, sont donnés par $c(x, u, t)$. La fonction $a(x, u, t)$ indique le point atteint à l'instant $t+1$ en partant d'un point x lorsqu'un contrôle u est exercé à l'instant t , cette fonction représente une loi de mouvement :

$$x_{t+1} = a(x_t, u_t, t)$$

L'équation de programmation dynamique est alors donnée par :

$$V(x, t) = \min_u (c(x, u, t) + V(a(x, u, t), t+1)) \quad (6.1)$$

avec, pour tout x , la condition terminale $V(x, h) = C_h(x)$, où $C_h(x)$ est un coût associé à un état x et à l'instant final h qui est fixe. Le coût de la trajectoire optimale est donc donné pour un horizon allant de t à h (de longueur $h-t$).

Pour mieux comprendre le principe de la programmation dynamique, appliquons cette méthode au problème de la recherche d'un chemin de moindre coût dans un graphe G . Pour chaque arc allant de x à y notons c_{xy} son coût et u_{xy} le contrôle qui permet de passer de x à y . Le contrôle u_{xx} correspond à un mouvement de coût nul ($c_{xx}=0$) allant de x à x . La loi de mouvement est alors donnée par

$$y = a(x, u_{xy}, t).$$

Elle ne dépend pas du temps t mais seulement de x et du contrôle u_{xy} . Les coûts sont définis par

$$c(x, u_{xy}, t) = c_{xy}.$$

On souhaite trouver le chemin de moindre coût d'un point s à un point p . On définit $V(x, t)$ comme étant le plus petit coût possible pour un chemin allant de x à p et de longueur $(h-t)$. On pose alors $V(x, h) = C_h(x) = \infty$ pour $x \neq p$ et $V(p, h) = C_h(p) = 0$. On obtient en utilisant (6.1) le coût du chemin recherché. Celui-ci est donné par $V(s, t)$ dès que pour tout point z , on a $V(z, t) = V(z, t+1)$. On peut montrer qu'il existe une valeur de t telle que cela est vérifié avec $h-t$ inférieur au nombre de sommets du graphe G . On prendra donc h égal au nombre de sommets du graphe G . Dans [BEL 1958] un algorithme identique est présenté pour la recherche d'un plus court chemin du sommet p aux autres dans un graphe orienté quelconque. Il est montré dans cet article que si cet algorithme ne trouve pas les plus courts chemins après un nombre d'itérations supérieur ou égal au nombre de sommets, c'est qu'il existe un circuit négatif dans ce graphe.

La figure 6.4 montre un exemple où le chemin de plus petit poids est calculé par programmation dynamique. On pose $h=4$ et donc on part de $V(.,4)$ connu avec $V(p,4)=0$, $V(x,4)=\infty$, $V(y,4)=\infty$ et $V(s,4)=\infty$. A partir de 3 itérations la solution n'évolue plus, on

obtient alors la longueur des chemins de plus petit poids partant de x , y ou s et allant vers p . Pour s le chemin de plus petit poids trouvé est $V(s,1)=4$. En calculant le minimum dans la formule (6.1), on trouve

$$V(s,1) = c_{sy} + V(a(s, u_{sy}, 1), 2) = 4$$

$$V(y,1) = c_{yx} + V(a(y, u_{yx}, 1), 2) = 3$$

$$V(x,1) = c_{xp} + V(a(x, u_{xp}, 1), 2) = 1$$

Donc les commandes u_{sy} , u_{yx} et u_{xp} sont optimales, on en déduit que le chemin de plus petit poids passe de s à y , puis de y à x , et de x à p .

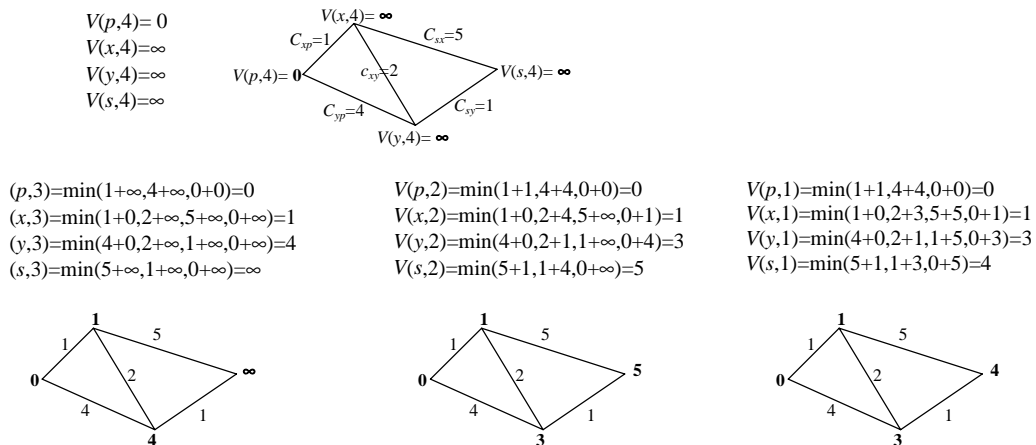


Figure 6.4 : Exemple de calcul du chemin de plus petit poids par programmation dynamique

Lorsqu'on applique la programmation dynamique à un système dont l'état évolue de façon Markovienne, on utilise une matrice de transition en guise d'équation de mouvement. On trouve alors en un point donné plusieurs trajectoires possibles qui, ensemble, donnent une espérance de coût optimal. A chaque itération, pour avoir un coût optimal, une commande est appliquée à chacun des taux de transition contrôlables. Dans le cas d'une ligne de production il s'agit des taux de service μ_i de chaque station exponentielle qui peut varier de 0 à $\bar{\mu}_i$. Le taux d'arrivée des demandes n'est pas contrôlable et il restera constant selon nos hypothèses. Si pour un état donné, une mise en marche d'une station accroît les coûts, alors la meilleure des commandes consiste à rendre nul le taux de service de cette station.

Nous avons considéré, comme dans [VEA 1994], deux types de coûts, tous deux sont calculés sur un horizon infini. Le premier est une somme des coûts instantanés avec pondération dans le temps par une exponentielle inverse décroissante. Le second est un coût instantané moyen. Le calcul du coût pour un horizon h est calculé à partir de celui d'un horizon $h-1$. L'horizon étant infini, le calcul du coût s'obtient alors en faisant tendre progressivement l'horizon vers l'infini. Une question importante consiste à vérifier si cette procédure permet d'obtenir une politique de contrôle stationnaire optimale. Dans [BER 1976] des conditions garantissant l'existence d'une politique de contrôle stationnaire optimale sont

données. Dans [ROS 1970], il est montré que le coût d'un contrôle stationnaire optimal est solution d'un point fixe et que cette solution s'obtient par des itérations successives qui optimisent le contrôle. On a donc une politique optimale stationnaire obtenue qui ne dépend pas du temps mais seulement de l'état courant.

L'évolution de l'état des systèmes considérés à un instant donné et les décisions prises sont uniquement fonctions de l'état du système à l'instant qui précède. Ce processus d'évolution d'un tel système est appelé processus de décision Markovien. Il peut s'effectuer soit à temps discret soit selon une évolution continue du temps. Nous ne considérerons ici que des temps continus. Lippman a montré en [LIP 1975] qu'il était possible de définir des périodes de décisions. Celles-ci correspondent au temps moyen d'arrivée des événements. Ce découpage du temps permet de faire des itérations comme si le temps était discret.

Dans une ligne de production telle que nous l'avons définie en section 6.1, le contrôle qui s'effectue sur le taux de service μ_i de chaque station i est fonction de X , le vecteur d'état global défini dans les chapitres précédents. Ce contrôle détermine la politique de gestion de production notée π . A une politique π correspondent des taux de production $\mu_i(X)$. La demande, qui arrive selon un taux λ , est le principal événement incontrôlable. On en déduit le générateur infinitésimal qui fait office d'équation de mouvement.

Nous montrons dans ce qui suit les calculs de la politique optimale pour un coût pondéré par une exponentielle inverse (voir la section 6.2.1) et pour un coût moyen (voir la section 6.2.2). Chacun de ces critères utilise un taux de coût instantané $c(X)$ qui est la somme des coûts de possession et des coûts de non satisfaction des demandes. Les coûts de possession sont proportionnels au nombre de produits présents dans le système et les coûts de non satisfaction de la demande sont proportionnels au nombre de demandes non servies à temps (les demandes ne sont jamais perdues). Les opérations de transformation appliquées aux produits le long d'une ligne de production peuvent changer le coût de possession, ainsi ce coût peut varier suivant l'emplacement des produits dans le système. Les calculs ci-après sont standards et plus de détails peuvent être trouvés dans les livres [BER 1995] et [PUT 1994].

6.2.1 Politique optimale pour un coût pondéré

Nous considérons ici la somme de coûts instantanés pondérés par une exponentielle inverse qui décroît dans le temps et tend vers zéro. La pente initiale de cette exponentielle inverse vaut $-\alpha$. Pour un paramètre α strictement positif, sachant que la politique est π et que l'état initial est X , la valeur de ce coût est la suivante :

$$V^{\pi, \alpha}(X) = E_{\pi} \left[\int_0^{\infty} e^{-\alpha t} c(X(t)) dt \mid X(0) = X \right] \text{ pour } \alpha > 0 \quad (6.2)$$

La fonction $E_{\pi} [C \mid X(0) = X]$ calcule l'espérance mathématique de C pour une politique π et sachant que l'état initial est X . On peut trouver une politique stationnaire optimale π^* dont

le coût pondéré est donné par la formule (6.3) qui doit être vérifiée pour tout état initial X .

On a :

$$V^{\pi^*,\alpha}(X) = \min_{\pi \in \Pi} V^{\pi,\alpha}(X) \quad (6.3)$$

avec Π l'ensemble des politiques admissibles. On peut atteindre cette valeur par programmation dynamique en calculant ce coût sur un horizon fini correspondant à l'occurrence de n événements discrets. En modifiant la formule (6.2) on trouve

$$V_n^{\pi,\alpha}(X) = E_\pi \left[\int_0^{\tau_n} e^{-\alpha t} c(X(t)) dt \mid X(0) = X \right] \quad (6.4)$$

avec τ_n l'instant d'occurrence du $n^{\text{ième}}$ événement discret. Pour un horizon fini la politique optimale peut correspondre à un contrôle non stationnaire. Cependant étant donnée l'évolution Markovienne du système, c'est-à-dire sans mémoire, on ne peut pas avoir de changement du contrôle tant qu'un événement ne s'est pas produit et que l'état du système n'a pas changé.

Le taux d'arrivée d'un événement quelconque est égal à la somme des taux d'arrivée de chaque événement possible. Pour chaque état, le temps moyen d'arrivée d'un événement quelconque n'est pas toujours le même car, par exemple, dans le cas d'une ligne de production, les taux de service varient d'un état à un autre en fonction du pilotage. Pour que ce taux d'arrivée d'un événement quelconque soit constant, Lippman [LIP 1975] a introduit des événements fictifs pour maintenir constante la somme des taux d'arrivée de chaque événement. Par exemple, dans une ligne de production, lorsque le taux $\mu_i(X)$ est inférieur à sa valeur maximale $\bar{\mu}_i$, il faut introduire un événement fictif dont le taux est $\bar{\mu}_i - \mu_i(X)$. L'instant d'arrivée de cet événement fictif ou d'un service selon un taux $\mu_i(X)$ est alors constant et de valeur $1/\bar{\mu}_i$. Lors de l'occurrence de cet événement fictif, l'état courant du système ne change pas et tout se passe s'il ne s'était rien produit.

Entre deux événements l'état du système reste inchangé, donc le coût instantané $c(X)$ et les taux $\mu_i(X)$ restent inchangés. A l'instant initial et jusqu'à l'arrivée d'un premier événement, l'état est connu et le coût instantané $c(X)$ l'est aussi. On peut alors écrire l'équation (6.4) de la façon suivante :

$$V_n^{\pi,\alpha}(X) = E_\pi \left[\int_0^{\tau_1} e^{-\alpha t} c(X) dt \right] + E_\pi \left[\int_{\tau_1}^{\tau_n} e^{-\alpha t} c(X(t)) dt \mid X(0) = X \right] \quad (6.5)$$

avec τ_n l'instant d'occurrence du $n^{\text{ième}}$ événement discret. En simplifiant on trouve :

$$V_n^{\pi,\alpha}(X) = \frac{1}{\Lambda + \alpha} \left\{ c(X) + \lambda V_{n-1}^{\pi,\alpha}(X + d_\lambda) + \sum_{i=1}^N (\mu_i(X) \cdot V_{n-1}^{\pi,\alpha}(X + d_{\mu_i})) + \sum_{i=1}^N ((\bar{\mu}_i - \mu_i(X)) \cdot V_{n-1}^{\pi,\alpha}(X)) \right\} \quad (6.6)$$

où d_{μ_i} et d_λ sont les variations de l'état générées respectivement par le service d'une pièce dans la station i et par l'arrivée d'une demande. On a aussi $\Lambda = \lambda + \sum_{i=1}^N \bar{\mu}_i$ et N le nombre

d'étages comportant une station monoserveur. En modifiant (6.6) on obtient alors l'équation de programmation dynamique suivante :

$$V_n^\alpha(X) = \frac{1}{\Lambda + \alpha} \left\{ c(X) + \lambda V_{n-1}^\alpha(X + d_\lambda) + \sum_{i=1}^N (\bar{\mu}_i \cdot V_{n-1}^\alpha(X)) + \sum_{i=1}^N \min_{0 \leq \mu_i \leq \bar{\mu}_i} (\mu_i \cdot (V_{n-1}^\alpha(X + d_{\mu_i}) - V_{n-1}^\alpha(X))) \right\} \quad (6.7)$$

Suivant le signe de $(V_{n-1}^\alpha(X + d_{\mu_i}) - V_{n-1}^\alpha(X))$ le taux μ_i sera soit nul soit égal à la valeur maximale $\bar{\mu}_i$. La politique optimale est donc de type "bang bang" [VEA 1994] car soit les stations ont un taux de production maximal soit elles sont à l'arrêt. Elle consiste à choisir $\mu_i(X, n) = 0$ si $(V_{n-1}^\alpha(X + d_{\mu_i}) - V_{n-1}^\alpha(X))$ est positif et $\mu_i(X, n) = \bar{\mu}_i$ sinon.

On peut donc modifier (6.7) de la façon suivante,

$$V_n^\alpha(X) = \frac{1}{\Lambda + \alpha} \left\{ c(X) + \lambda V_{n-1}^\alpha(X + d_\lambda) + \sum_{i=1}^N \bar{\mu}_i \cdot \min(V_{n-1}^\alpha(X + d_{\mu_i}), V_{n-1}^\alpha(X)) \right\} \quad (6.8)$$

Le terme de droite en (6.8) est fonction de V_{n-1}^α et de X , il est donc possible de définir un opérateur T fonction de ces deux éléments pour remplacer ce terme. Ainsi l'égalité (6.8) peut être notée plus simplement par l'écriture (6.9) qui suit :

$$V_n^\alpha(X) = T(V_{n-1}^\alpha)(X) \quad (6.9)$$

A partir d'un certain nombre d'itérations le contrôle devient stationnaire et on obtient ainsi la politique optimale pour un horizon infini. On a

$$V_{n-1}^{\pi^*, \alpha}(X) = \lim_{n \rightarrow \infty} V_n^\alpha(X) \quad (6.10)$$

Lorsque cette limite existe, $V^{\pi^*, \alpha}$ est le point fixe de T , on a ainsi

$$V^{\pi^*, \alpha}(X) = T(V^{\pi^*, \alpha})(X) \quad (6.11)$$

Le calcul numérique de cette politique optimale et de son coût nécessite à chaque itération la détermination de $V_n^\alpha(X)$ pour chaque état X . Or l'espace d'état est infini puisque le nombre de demandes en attente n'est pas limité. Il est donc nécessaire de tronquer l'espace d'état pour qu'il soit fini et que le nombre d'états ne soit pas trop grand.

Considérons le cas d'une ligne de deux étages, on a $X \in \mathbb{N} \times \mathbb{Z}$, il est donc nécessaire de limiter le nombre de produits intermédiaires, de produits finis et demandes non servies. Limitons par exemple ces quantités à trente. Ainsi la valeur de X_1 va de 0 à 30 et la valeur de X_2 va de -30 à 30, et on obtient avec ces hypothèses plus de 1800 états.

Pour ne pas quitter ce domaine réduit, toute transition permettant d'en sortir aura donc un taux de service nul. Ainsi dans l'exemple considéré, $\mu_2(X)$ est nul pour tout X tel que $X_2 \geq 30$, $\mu_1(X)$ est nul pour tout X tel que $X_1 \geq 30$ et le taux d'arrivée $\lambda(X)$ des clients qui normalement est constant doit s'annuler si $X_2 \leq -30$. Cette diminution de l'espace d'états génère des effets de bord qui diminuent la précision des résultats. La taille de l'espace doit donc permettre de faire un compromis entre la précision et la durée des calculs.

Le calcul du coût pondéré s'obtient facilement par des méthodes utilisant la programmation dynamique. Le coût pondéré est plus difficile à interpréter et moins utilisé que le coût moyen. En particulier le coût pondéré dépend de l'état initial, alors qu'en général, ce n'est pas le cas pour le coût moyen. Ce dernier est l'objet de la section qui suit.

6.2.2 Politique optimale pour un coût moyen

La valeur du coût moyen pour une politique π ne dépend pas de l'état initial lorsqu'il n'y a qu'une seule classe d'états ergodiques. Il correspond à l'espérance sur un horizon infini, de la moyenne des coûts instantanés. Il est donné par la formule suivante :

$$g^\pi = \lim_{T \rightarrow \infty} E_\pi \left[\frac{\int_0^T c(X(t)) dt}{T} \right] \quad (6.12)$$

La fonction E_π représente l'espérance mathématique pour politique π . Pour qu'une politique stationnaire soit optimale, il suffit que son coût moyen (6.12) soit inférieur à celui de toute autre politique admissible.

Lorsque l'horizon n'est pas infini on peut calculer, pour une politique π donnée, le cumul des coûts instantanés. Pour un horizon fini correspondant à l'occurrence de n événements discrets, avec un état initial X et une politique π on définit :

$$V_n^\pi(X) = E_\pi \left[\int_0^{\tau_n} c(X(t)) dt \mid X(0) = X \right] \quad (6.13)$$

qui correspond à l'espérance du coût pour un horizon se terminant à τ_n l'instant d'arrivée du $n^{\text{ième}}$ événement discret. Lorsque n tend vers l'infini alors le cumul des coûts devient lui aussi infini. Calculer $V_n^\pi(X)$ pour n très grand et diviser par τ_n n'est pas convenable car cela nécessite l'addition de petits nombres à d'autres très grands. On utilise alors une valeur qui correspond, pour n très grand, à la différence transitoire de coût due à une différence d'état initial. Cette différence est définie par :

$$h_n^\pi(X) = V_n^\pi(X) - V_n^\pi(s) \quad (6.14)$$

avec s un état quelconque qui ne change pas quel que soit n . Pour une politique donnée si le temps moyen entre deux événements est $1/\Lambda$ alors il est possible d'exprimer le coût moyen entre le $n^{\text{ième}}$ et le $(n+1)^{\text{ième}}$ événement par :

$$g_n^\pi / \Lambda = (V_{n+1}^\pi(s) - V_n^\pi(s)) \quad (6.15)$$

La probabilité de l'état après le $n^{\text{ième}}$ événement tend vers une probabilité stationnaire donc g^π est la limite de g_n^π pour n infini. On peut évaluer cette limite en itérant les formules suivantes :

$$\begin{aligned}
h_0^\pi(X) &= 0 \\
h_{n+1}^\pi(s) &= 0 \\
g_{n+1}^\pi + \Lambda h_{n+1}^\pi(X) &= c(X) + \lambda h_n^\pi(X + d_\lambda) \\
&\quad + \sum_{i=1}^N (\mu_i(X) \cdot h_n^\pi(X + d_{\mu_i})) + \sum_{i=1}^N ((\bar{\mu}_i - \mu_i(X)) \cdot h_n^\pi(X))
\end{aligned} \tag{6.16}$$

On a $\Lambda = \lambda + \mu_1 + \dots + \mu_N$ car comme pour le calcul du coût pondéré on introduit des événements fictifs pour rendre le taux d'arrivée des événements constant quel que soit le contrôle appliqué. En modifiant (6.16) on obtient alors les équations de programmation dynamique suivantes :

$$\begin{aligned}
h_0(X) &= 0 \\
h_{n+1}(s) &= 0 \\
g_{n+1} + \Lambda h_{n+1}(X) &= c(X) + \lambda h_n(X + d_\lambda) \\
&\quad + \sum_{i=1}^N \left(\min_{0 \leq \mu_i \leq \bar{\mu}_i} (\mu_i \cdot (h_n(X + d_{\mu_i}) - h_n(X))) \right) + \sum_{i=1}^N (\bar{\mu}_i \cdot h_n(X))
\end{aligned} \tag{6.17}$$

La politique optimale est de type "bang bang" et elle consiste à choisir $\mu_i(X, n) = 0$ si $(h_{n-1}(X + d_{\mu_i}) - h_{n-1}(X))$ est positif et $\mu_i(X, n) = \bar{\mu}_i$ sinon. A chaque itération la politique de contrôle est optimisée. Le système d'équations (6.17) peut être simplifié de la façon suivante :

$$\begin{aligned}
h_0(X) &= 0 \\
h_{n+1}(s) &= 0 \\
\frac{g_{n+1}}{\Lambda} + h_{n+1}(X) &= \frac{c(X) + \lambda h_n(X + d_\lambda) + \sum_{i=1}^N (\bar{\mu}_i \cdot \min(h_n(X + d_{\mu_i}), h_n(X)))}{\Lambda}
\end{aligned} \tag{6.18}$$

On définit un opérateur T fonction de h_n et de X par le terme de droite de l'équation (6.18).

On obtient alors le système d'équations qui suit ((6.18) et (6.20) sont identiques).

$$\begin{aligned}
h_0(X) &= 0 \\
\frac{g_{n+1}}{\Lambda} &= T(h_n)(s)
\end{aligned} \tag{6.19}$$

$$\frac{g_{n+1}}{\Lambda} + h_{n+1}(X) = T(h_n)(X) \tag{6.20}$$

A partir d'un certain nombre d'itérations le contrôle calculé devient stationnaire et on obtient ainsi la politique optimale pour un horizon infini. Avec $h(s) = 0$, on tend à vérifier l'équation d'optimalité suivante :

$$h(X) + \frac{g^{\pi^*}}{\Lambda} = T(h)(X) \tag{6.21}$$

Cette équation d'optimalité est un résultat standard [BER 1976], [ROS 1970] ou [VER 2000]. Pour le calcul itératif de la politique optimale, une méthode légèrement différente est présentée en [ROS 1970]. Cette méthode consiste à calculer le coût moyen sur un horizon

infini pour une politique π_k donnée, on obtient alors g^{π_k} et h^{π_k} comme solutions du système d'équation (6.16) pour $n = \infty$. La politique π_{k+1} est ensuite définie comme celle qui permet de minimiser $T(h^{\pi_k})(X)$. L'itération se poursuit alors jusqu'à ce que $\pi_{k+1} = \pi_k$. Un calcul itératif qui utilise les équations (6.19) et (6.20) est moins précis mais plus rapide car il évite la résolution d'un système linéaire à chaque itération ; il est utilisé en [VEA 1994].

Comme pour le coût pondéré il faut limiter artificiellement le nombre d'états pour rendre possibles les calculs. Ainsi le nombre d'opérations nécessaires dans ces calculs nous interdit l'étude de systèmes de grande taille. Cette méthode est cependant la seule qui est efficace dans le calcul numérique d'une politique optimale.

On peut aussi en utilisant la programmation dynamique montrer certaines propriétés qui caractérisent les politiques optimales.

6.3 Caractérisation partielle des politiques.

La programmation dynamique permet d'approcher numériquement la politique optimale d'un système de production mais parfois elle permet aussi de démontrer qu'une certaine catégorie de contrôle est optimale ou non optimale. On peut ainsi obtenir une caractérisation partielle des politiques optimales.

Par exemple, il est montré dans [VEA 1992] et [WEB 1987] que la commande optimale d'un système de production vérifie des propriétés de monotonie. Soit D un ensemble de vecteurs de transition qui sont chacun associés à l'occurrence d'un événement. Lorsqu'un événement se produit la modification du vecteur d'état X se fait en ajoutant le vecteur de transition associé. On dit qu'un contrôle qui est défini en fonction du vecteur d'état par $\mu_i(X)$ est D -monotone si :

$$\mu_j(X) \leq \mu_j(X + d) \quad (6.22)$$

pour tout taux de transition μ_j contrôlable et pour tout vecteur de transition $d \in D$ telle que $d \neq d_{\mu_j}$, où d_{μ_j} est le vecteur de transition associé à l'occurrence de l'événement « service d'un produit à la station monoserveur j » selon un taux μ_j .

Dans [VEA 1992] il est montré que, sous certaines hypothèses et pour certains modèles de type « file d'attente », un contrôle optimal doit être D -monotone. Pour une ligne de production à deux mailles, on a un contrôle optimal D -monotone avec $D = \{d_{\mu_1}, d_{\mu_2}, d_\lambda\}$. Cette propriété signifie qu'en tout point X la fonction de contrôle $\mu_i(X)$ sera plus petite que $\mu_i(X + d)$ et plus grande que $\mu_i(X - d)$ avec d un vecteur se composant d'une somme quelconque des vecteurs de $D \setminus \{d_{\mu_i}\}$ (se montre facilement à partir de (6.22)).

Puisque la politique optimale est de type « bang-bang » deux cas se produisent, soit $\mu_i(X) = \bar{\mu}_i$ et alors $\mu_i(X) \leq \mu_i(X + d)$ ce qui implique $\mu_i(X + d) = \bar{\mu}_i$, soit

$\mu_i(X) = 0$ et alors $\mu_i(X - d) \leq \mu_i(X)$ ce qui implique $\mu_i(X - d) = 0$. Par récurrence on montre que si $\mu_i(X) = \bar{\mu}_i$ (respectivement $\mu_i(X) = 0$) alors $\mu_i\left(X + \sum_j \alpha_j d_j\right) = \bar{\mu}_i$ (respectivement $\mu_i\left(X - \sum_j \alpha_j d_j\right) = 0$) avec $\alpha_j \in \mathbb{N}$ et $d_j \in D \setminus \{d_{\mu_i}\}$. Ainsi pour tout état global X si le taux de service de la station i vaut $\bar{\mu}_i$ (respectivement zéro) alors ce taux reste inchangé dans le cône $C_i(X)$ (respectivement $C_i^-(X)$) défini par :

$$C_i(X) = \left\{ X + \sum_j \alpha_j d_j \mid \alpha_j \in \mathbb{N}; d_j \in D \setminus \{d_{\mu_i}\} \right\}$$

$$C_i^-(X) = \left\{ X - \sum_j \alpha_j d_j \mid \alpha_j \in \mathbb{N}; d_j \in D \setminus \{d_{\mu_i}\} \right\}$$

Les cônes $C_i(X)$ et $C_i^-(X)$ sont orientés dans des directions opposées. Si en X la station i marche et si elle est à l'arrêt en $X + d_{\mu_i}$ alors elle marche pour tout état de $C_i(X)$ et elle est aussi à l'arrêt pour tout état de $C_i^-(X + d_{\mu_i})$. En dehors de ces deux cônes le contrôle de la station i peut prendre plusieurs valeurs. La figure 6.5 montre pour une ligne de deux étages des cônes C_1, C_1^-, C_2 et C_2^- en différents points de l'espace d'état. Un agrandissement permet de voir comment, en des points X et $X + \mu_1$, sont respectivement disposés les cônes C_1 et C_1^- associés à la marche et à l'arrêt de la station 1. On peut montrer que le contrôle des stations est composé d'une zone de marche et d'une zone d'arrêt qui sont délimitées par une courbe dite de commande. Pour chaque point X et $X + \mu_i$ entre lesquels la courbe de commande d'une station i passe, on a deux cônes $C_i(X)$ et $C_i^-(X + d_{\mu_i})$ séparés par cette même courbe de commande. Ceci est illustré par sur figure 6.5.

Les zones correspondant à la marche d'une station ont pour intersection un ensemble où toutes les stations fonctionnent. L'ensemble des états de cette intersection sont ergodiques. Cependant, certaines transitions permettent de sortir de cette zone d'intersection, des états ergodiques existent donc à l'extérieur de cette zone. L'ensemble de tous les états ergodiques est en fait défini par l'ensemble des vecteurs d'état X vérifiant $\mu_i(X) = \bar{\mu}_i$ pour tout i ou vérifiant $\mu_i(X - d_{\mu_i}) = \bar{\mu}_i$ pour tout i . Ils se trouvent donc soit dans la zone où toutes les stations marchent soit en bordure de cette zone.

Un tel contrôle optimal D-monotone existe pour des lignes de plusieurs mailles [VEA 1992]. Des applications à un système avec assemblage ou désassemblage sont aussi possibles. Un système de production comportant une station d'assemblage est étudié en section 6.7.

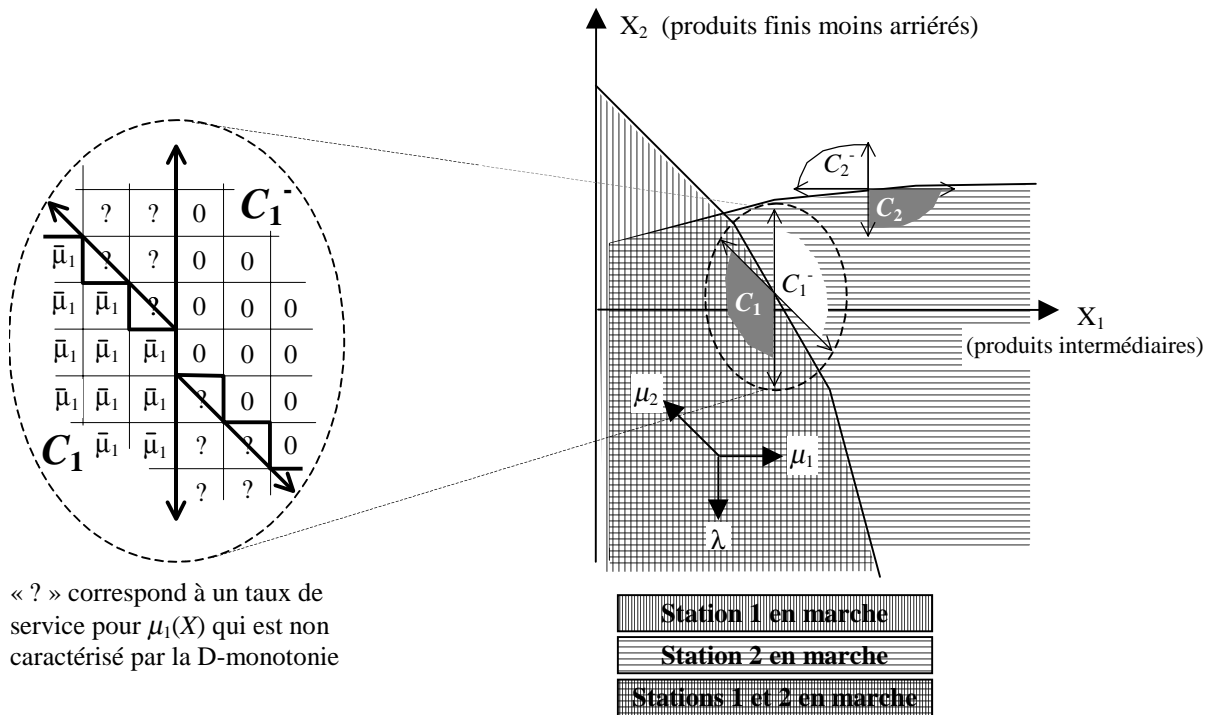


Figure 6.5 : Courbes de commande des stations 1 et 2 pour une ligne de deux étages avec représentation de cônes pour illustrer la D-monotonie du contrôle.

Dans [VEA 1994] des recherches plus détaillées sont faites sur la caractérisation des politiques optimales pour des lignes de deux mailles avec stations exponentielles, demande poissonnienne et un coût instantané $c(X) = X_1 + hX_2^+ + bX_2^-$. On y trouve les résultats suivants :

- Il est montré que dans ce cas une politique basestock ne peut jamais être optimale. La preuve est donnée par une technique de processus couplés.
- Avec certaines conditions sur les taux de coût et de service, il est possible de montrer qu'une politique maintenant des stocks nuls est optimale pour un coût pondéré.
- Pour un taux de production de la station 2 suffisamment élevé, la politique optimale est un contrôle où, lorsque X est tel que $X_2 < 0$ alors $\mu_2(X) = \bar{\mu}_2$ et sinon $\mu_2(X) = 0$. Ce contrôle consiste donc à produire au niveau de la station 2 si, et seulement si, il y a des demandes en attente.
- Pour $h < 1$, il est optimal de toujours avoir la station 2 en marche lorsque des produits intermédiaires sont disponibles. Ce résultat est étendu au cas $h = 1$ dans [KAR 2000].
- Si la courbe de commande optimale de la station 1 est inférieure à l'axe X_1 alors il est optimal de toujours produire dans la station 2 dans chaque état ergodique.

Plusieurs techniques existent pour obtenir ce type de caractérisation. On peut utiliser des processus couplés, pour analyser la différence de coût existant lorsque les états initiaux sont

différents. On peut aussi utiliser les équations de la programmation dynamique pour effectuer une démonstration par récurrence. Ce type de caractérisation est difficile à obtenir. Pour se faire une idée sur les caractérisations possibles des politiques optimales d'un système il est intéressant de les calculer numériquement.

Dans les sections qui suivent nous calculons numériquement le coût des politiques optimales principalement pour faire des comparaisons avec des politiques sous-optimales. Les politiques sous-optimales considérées sont le basestock, le kanban, le kanban étendu et le kanban généralisé.

6.4 Comparaisons entre des politiques sous-optimales et la politique optimale pour une ligne de deux étages.

Le système étudié ici est une ligne de deux étages. La figure 6.6 illustre les principaux éléments de ce système. Il y a dans chaque étage i un processus de fabrication comportant une station monoserveur M_i à service exponentiel dont le taux de service μ_i vaut $\bar{\mu}_i$ pour la marche et zéro pour l'arrêt. Les demandes arrivent selon un processus poissonien de taux λ . Chaque demande est satisfaite s'il existe des produits finis, sinon elle est mise en attente en D_2 . Le stock de produits bruts P_0' est supposé n'être jamais vide. La file d'attente P_1' contient des produits intermédiaires. Pour les lignes de production des figures 6.7, 6.8 et 6.9, P_1' représente le stock de produits intermédiaires P_1 plus le stock tampon du processus de fabrication F_2 , conformément aux notations de la section 3.1.

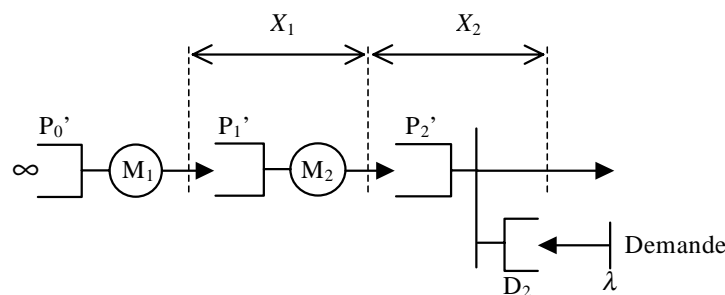


Figure 6.6 : Une ligne de deux mailles avec pilotage des processus de fabrication

Par hypothèse, le coût instantané de possession d'un produit intermédiaire (entre la sortie des deux stations) est un taux de valeur 1 ; pour un produit fini ce taux est h , et pour une demande non servie il y a un coût de non satisfaction de la demande dont le taux est b . En P_0' et M_1 le coût est nul. Le coût instantané correspond donc à la fonction suivante :

$$c(X) = X_1 + hX_2^+ + bX_2^- \quad (6.23)$$

Le vecteur X est un état global et on peut rappeler que la composante X_1 correspond aux produits intermédiaires, X_2^+ , la valeur positive de X_2 , correspond aux produits finis, et X_2^- ,

la valeur négative de X_2 , correspond aux demandes non servies. A partir de ces données initiales, on peut calculer par programmation dynamique la politique optimale pour un coût pondéré ou moyen. Les équations de programmation dynamique utilisées sont (6.8) pour un coût pondéré et (6.18) pour un coût moyen avec $X = (X_1, X_2)$, $d_\lambda = (0, -1)$, $d_{\mu_1} = (1, 0)$ et $d_{\mu_2} = (-1, 1)$.

Pour calculer le coût des politiques sous-optimales, nous allons reprendre les techniques de programmation dynamique utilisées pour le calcul du coût des politiques optimales, ainsi les erreurs numériques seront de même nature et les résultats pourront être comparés. Contrairement à une politique optimale où l'on calcule la courbe de commande et le coût de façon simultanée, le coût d'une politique sous optimale ne sera calculé qu'après avoir fixé le profil de la courbe de commande équivalente.

Les politiques sous-optimales étudiées sont le basestock, le kanban et le kanban généralisé (voir figures 6.7, 6.8 et 6.9). Nous n'avons pas étudié le kanban étendu car il est montré dans le chapitre précédent que cette politique est un cas particulier de la politique kanban généralisé lorsqu'on a un système de production avec deux étages en série et une station monoserveur par étage.

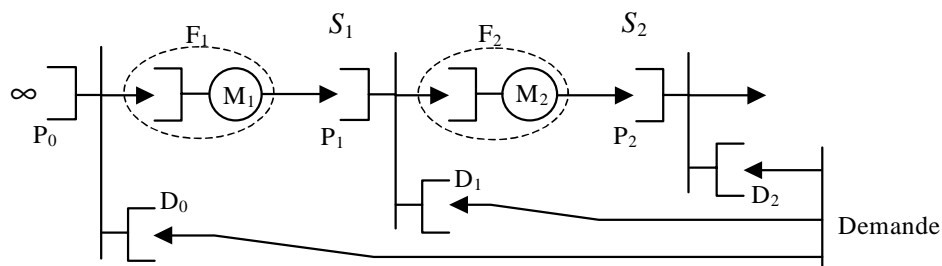


Figure 6.7 : Basestock sur une ligne de deux mailles

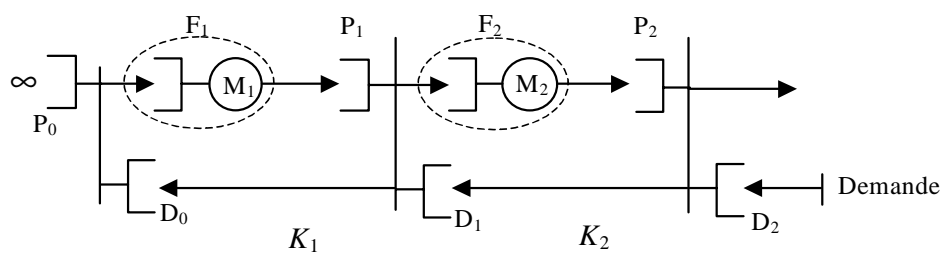


Figure 6.8 : Kanban sur une ligne de deux mailles

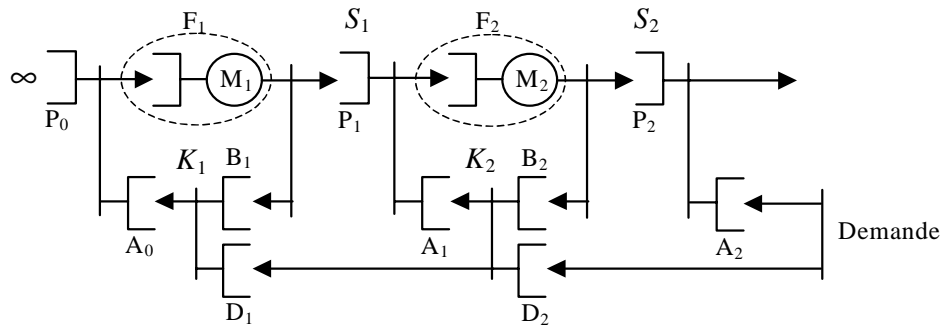


Figure 6.9 : Kanban généralisé sur une ligne de deux mailles

Pour que les coûts d'une politique basestock, kanban ou kanban généralisé puissent être comparés il faut déterminer pour chacun le paramétrage qui minimise le coût (K_i et S_i optimaux). Pour optimiser ces paramétrages une heuristique a été employée. Celle-ci consiste à regarder si localement le paramétrage utilisé pour la politique est optimal, lorsque celui-ci ne l'est pas, on définit une direction de descente qui indique le meilleur paramétrage. On fait évoluer le paramétrage dans cette direction tant que celle-ci permet de réduire le coût, sinon on recommence la procédure en vérifiant si localement le paramétrage est optimal. La procédure s'arrête quand un minimum local est atteint.

Considérons un exemple où la station 2 est fortement goulet ($\bar{\mu}_1 = 10$, $\bar{\mu}_2 = 1.1$). Les courbes de commande de la station 1 correspondant à cet exemple sont données figure 6.10 pour la politique optimale et des politiques sous-optimales (kanban, basestock et kanban généralisé). La station 1 ne fonctionne que pour les états situés en dessous de sa courbe de commande. Une comparaison visuelle des courbes de commande montre que c'est la courbe du kanban généralisé qui se situe à la plus faible distance de la courbe optimale. On observe aussi que la courbe de commande du basestock se situe par rapport aux autres courbes à une distance de la courbe optimale qui est la plus grande.

Dans le tableau 6.1, on a indiqué le surcoût par rapport à la politique optimale pour le kanban, le basestock et le kanban généralisé. Plusieurs taux $\bar{\mu}_1$ et $\bar{\mu}_2$ sont considérés. On remarque que le kanban généralisé est toujours la moins coûteuse des politiques sous-optimales étudiées. La supériorité du kanban généralisé est prévisible car il est toujours possible de rendre cette politique équivalente au basestock ou au kanban.

Rappelons que pour la structure étudiée dans cette section la politique kanban étendu est incluse dans la politique kanban généralisé au sens défini dans le chapitre précédent (voir section 5.2). Par conséquent le coût obtenu avec une politique kanban étendu (après optimisation des paramètres) est supérieur ou égal au coût obtenu avec une politique kanban généralisé (après optimisation des paramètres). Plus précisément ils seraient identiques dans le cas où les paramètres optimaux du kanban généralisé vérifient $K_2 \geq S_2$ mais le kanban généralisé sera meilleur lorsque son paramétrage optimal sera tel que $K_2 < S_2$.

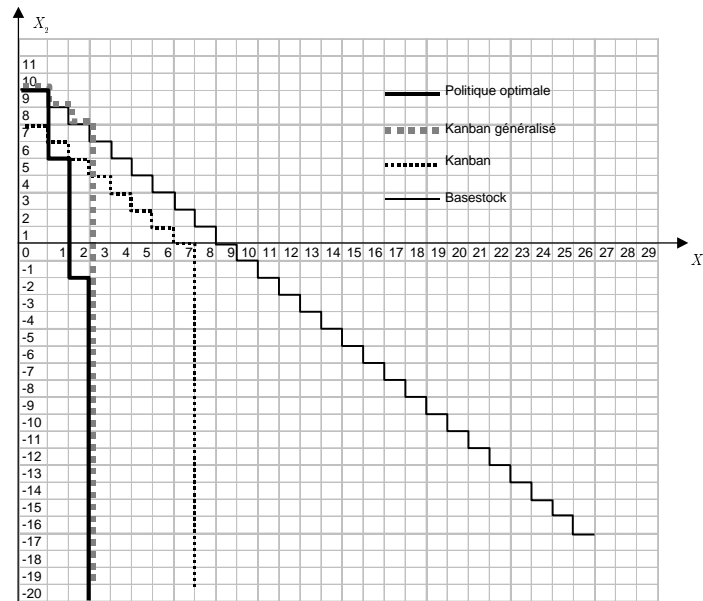


Figure 6.10 : Un exemple de courbes de commande de la station 1 pour une ligne de deux mailles, $\bar{\mu}_1 = 10$, $\bar{\mu}_2 = 1.1$ et pour différentes politiques

Si l'on considère les 5 cas présentés sur le tableau 6.1, tous les paramétrages optimaux du kanban généralisé vérifient $K_2 \geq S_2$ sauf dans celui où $\bar{\mu}_1 = 4$ et $\bar{\mu}_2 = 1.1$. Ainsi pour ce dernier cas dans lequel on a $K_2 < S_2$, le plus petit coût obtenu avec une politique kanban étendu est supérieur au meilleur coût obtenu avec une politique kanban généralisé alors que pour les autres cas il est égal.

	Surcoût par rapport au coût moyen de la politique optimale				
Données initiales ($\lambda = 1; h = 2; b = 4$)	$\bar{\mu}_1 = 1.2$ $\bar{\mu}_2 = 1.2$	$\bar{\mu}_1 = 2$ $\bar{\mu}_2 = 1.2$	$\bar{\mu}_1 = 1.2$ $\bar{\mu}_2 = 2$	$\bar{\mu}_1 = 2$ $\bar{\mu}_2 = 2$	$\bar{\mu}_1 = 4$ $\bar{\mu}_2 = 1.1$
Coût politique optimale	21,48	14,74	10,76	5,25	20,04
Surcoût basestock	0,36%	10,34%	3,29%	1,10%	14,61%
(S_1, S_2) optimal	(4,8)	(0,6)	(6,2)	(1,2)	(0,8)
Surcoût kanban	2,93%	3,16%	4,25%	5,62%	12,53%
(K_1, K_2) optimal	(6,8)	(1,5)	(7,2)	(1,2)	(1,6)
Surcoût kanban généralisé	0,36%	1,95%	3,29%	0,90%	1,87%
$(S_1, S_2, K_2, K_1=1)$ optimal	(4,8,15)	(0,6,6)	(6,2,12)	(1,2,5)	(0,9,4)

Tableau 6.1: Surcoût/ l'optimal pour différents taux de service

Comme il n'y a pas de coût associé aux pièces en attente dans le processus de fabrication F_1 , alors le paramètre K_1 d'un kanban généralisé pour une ligne de deux étages n'a pas d'influence sur le pilotage des stations monoserveur et sur la valeur du vecteur d'état qui

permet de calculer les coûts. On peut donc fixer de façon arbitraire la valeur de ce paramètre K_1 (par exemple à $K_1=1$) car il n'a pas d'influence dans le calcul des coûts.

Il est intéressant de constater que le surcoût du kanban généralisé par rapport à l'optimal n'excède pas 3% dans la majorité des cas. Les autres politiques peuvent atteindre des pourcentages bien plus élevés. On observe que le basestock est souvent très performant mais dans certaines situations, comme pour un effet goulet sur la machine aval ($\bar{\mu}_1 = 2$ et $\bar{\mu}_2 = 1.2$), cette politique peut devenir la plus coûteuse de celles étudiées ici. Les performances obtenues pour le kanban sont moyennes, mais rarement mauvaises. Le tableau 6.2 montre des statistiques sur les résultats du tableau 6.1.

Surcoût/optimal:	Basestock	Kanban	Kanban Généralisé
Maximum	14.61%	12.53%	3.29%
Moyen	5.94%	5.70%	1.67%
Minimum	0.36%	2.93%	0.36%
Ecart type	6.25%	3.96%	1.12%

Tableau 6.2: Maximum, moyenne et minimum du surcoût/ l'optimal

Les articles [KAR 2000] et [VEA 1994] contiennent tous deux ce type d'analyse de performance. Veatch et Wein n'ont pas traité le cas du kanban généralisé alors que cela est fait dans l'article de Karaesmen et Dallery. Ainsi nous avons pu vérifier l'exactitude d'une partie de nos calculs. De petites différences existent, celles-ci sont dues aux effets de bord et à la dimension choisie pour l'espace d'état tronqué. Augmenter l'espace d'état permet d'augmenter la précision des calculs cependant cela augmente aussi les temps de calcul.

Dans les sections qui suivent nous étudions des cas où un délai de satisfaction de la demande autorise un certain retard moyen.

6.5 Comparaisons pour une ligne de deux mailles avec délai

Nous étudions ici des lignes de deux mailles où il n'y a pas de coût de non satisfaction des demandes si le temps pour satisfaire la demande est inférieur à un certain délai. Dans l'industrie, cette hypothèse est tout à fait réaliste puisque qu'il est fréquent que les donneurs d'ordres donnent un délai à leurs fournisseurs.

On remarque que les vecteurs d'état global comportent une composante supplémentaire X_3 représentant le nombre de demandes non immédiatement satisfaites mais qui ne sont pas en retard. La composante X_2 représente les produits finis avec X_2^+ lorsqu'elle est positive et les demandes en retard avec X_2^- lorsqu'elle est négative. Les demandes en retard sont celles qui n'ont pas été servies après un temps supérieur au délai autorisé. Un coût (selon un taux b) de

non satisfaction de la demande est associé aux X_2^- demandes en retard ; par contre les X_3 demandes non encore en retard ne génèrent pas de coûts. Avec cette nouvelle interprétation de X_2^- , on utilise la fonction coût instantané $c(X) = X_1 + hX_2^+ + bX_2^-$ précédemment utilisée (6.23).

6.5.1 Cas d'un livraison au plus tôt avec délai

6.5.1.1 Description du système étudié

Pour ce cas, les demandes sont servies au plus tôt même si le délai autorise un retard supplémentaire. Le réseau de Petri de la figure 6.11 montre comment chaque demande, qui arrive par un processus poissonien de taux λ , devient en retard, après un délai distribué selon une loi exponentielle de taux $\lambda_{\text{délai}}$. Un arc inhibiteur donne la priorité à la satisfaction des demandes en retard. Lorsqu'il n'y a pas de demande en retard chaque demande est servie au plus tôt. Ainsi il n'est pas possible d'avoir à la fois des produits finis et des demandes non servies, donc X_3 est toujours nul lorsque X_2 est positif.

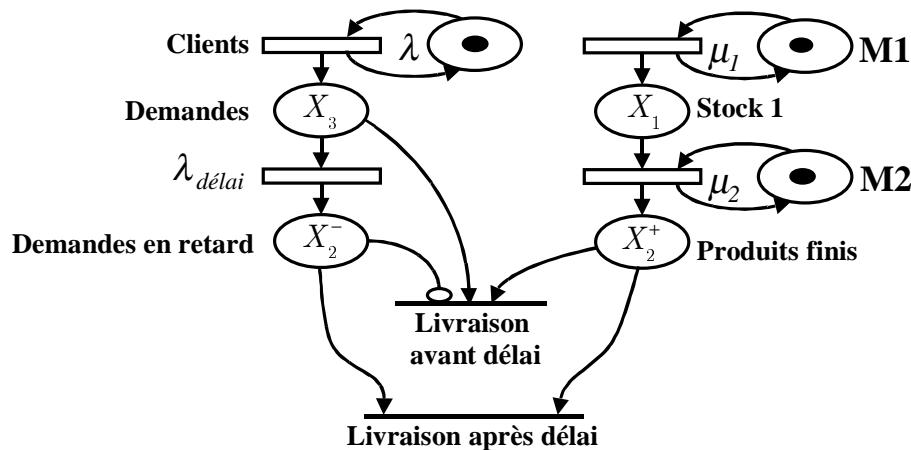


Figure 6.11 : Modèle RdP d'une ligne composée de deux stations M_1 et M_2 autorisant un délai de livraison

6.5.1.2 Politique optimale

En utilisant les équations de programmation dynamique (6.8) et (6.18), on peut calculer respectivement le coût pondéré et le coût moyen. Suivant l'état du système défini par le vecteur $X = (X_1, X_2, X_3)$, ces équations varient en fonction des transitions possibles.

- Si $X_2 > 0$ alors $d_{\mu_1} = (1, 0, 0)$, $d_{\mu_2} = (-1, 1, 0)$, $d_{\lambda} = (0, -1, 0)$ et aucun délai de taux $\lambda_{\text{délai}}$ ne peut exister car $X_3 = 0$.
- Si $X_2 = 0$ et $X_3 = 0$ alors $d_{\mu_1} = (1, 0, 0)$, $d_{\mu_2} = (-1, 1, 0)$ et $d_{\lambda} = (0, 0, 1)$.

- Si $X_2 = 0$ et $X_3 > 0$ on a alors $d_{\mu_1} = (1, 0, 0)$, $d_{\mu_2} = (-1, 0, -1)$, $d_{\lambda} = (0, 0, 1)$ et $d_{\lambda_{\text{délai}}} = (0, -1, -1)$.
- Si $X_2 < 0$ et $X_3 = 0$ alors $d_{\mu_1} = (1, 0, 0)$, $d_{\mu_2} = (-1, 1, 0)$ et $d_{\lambda} = (0, 0, 1)$.
- Si $X_2 < 0$ et $X_3 > 0$ alors $d_{\mu_1} = (1, 0, 0)$, $d_{\mu_2} = (-1, 1, 0)$, $d_{\lambda} = (0, 0, 1)$ et $d_{\lambda_{\text{délai}}} = (0, -1, -1)$.
- Pour $X_2 \leq 0$ et $X_3 > 0$, il peut se produire X_3 transitions de direction $d_{\lambda_{\text{délai}}}$. Pour chaque demande en attente le délai s'écoule avec un taux $\lambda_{\text{délai}}$, donc lorsqu'il y en a X_3 , le temps entre l'écoulement de deux délais suit une loi exponentielle de taux $X_3 \cdot \lambda_{\text{délai}}$.

La politique optimale de ce système a été étudiée en [FEN 1997] pour le coût pondéré. Nous avons complété l'étude par le calcul de la politique optimale pour le coût moyen.

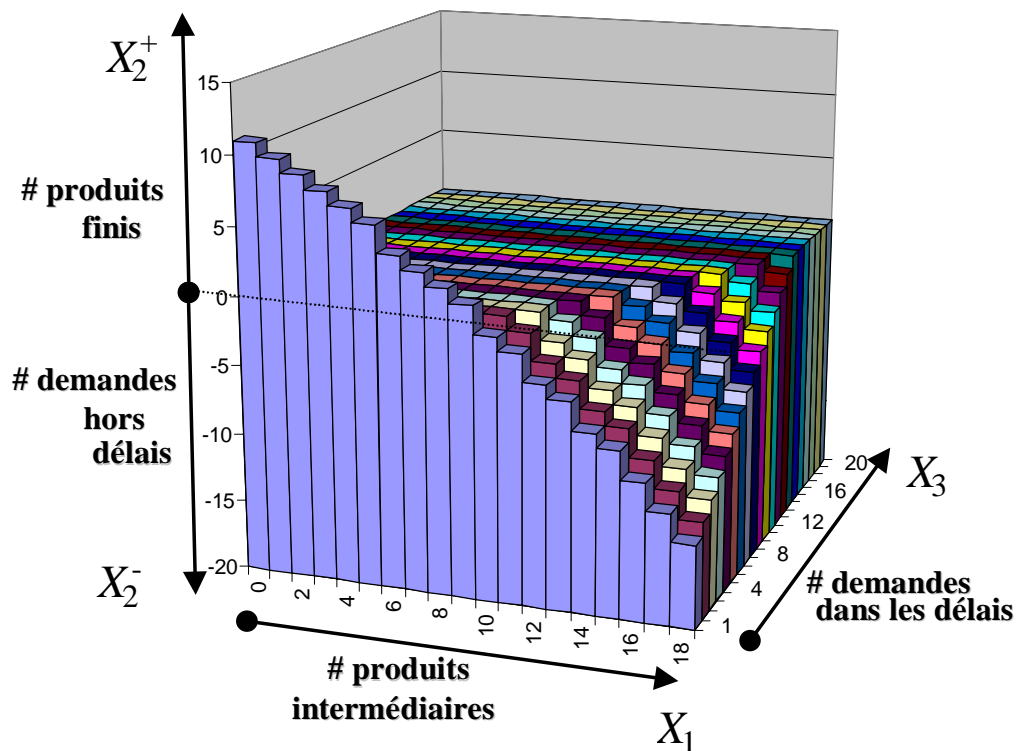


Figure 6.12 : Courbe de commande optimale de la station 1 avec $\mu_1=1.2$, $\mu_2=1.2$, $\lambda=1$ et $\lambda_{\text{délai}}=1$, c'est-à-dire un délai moyen de 1 unité de temps

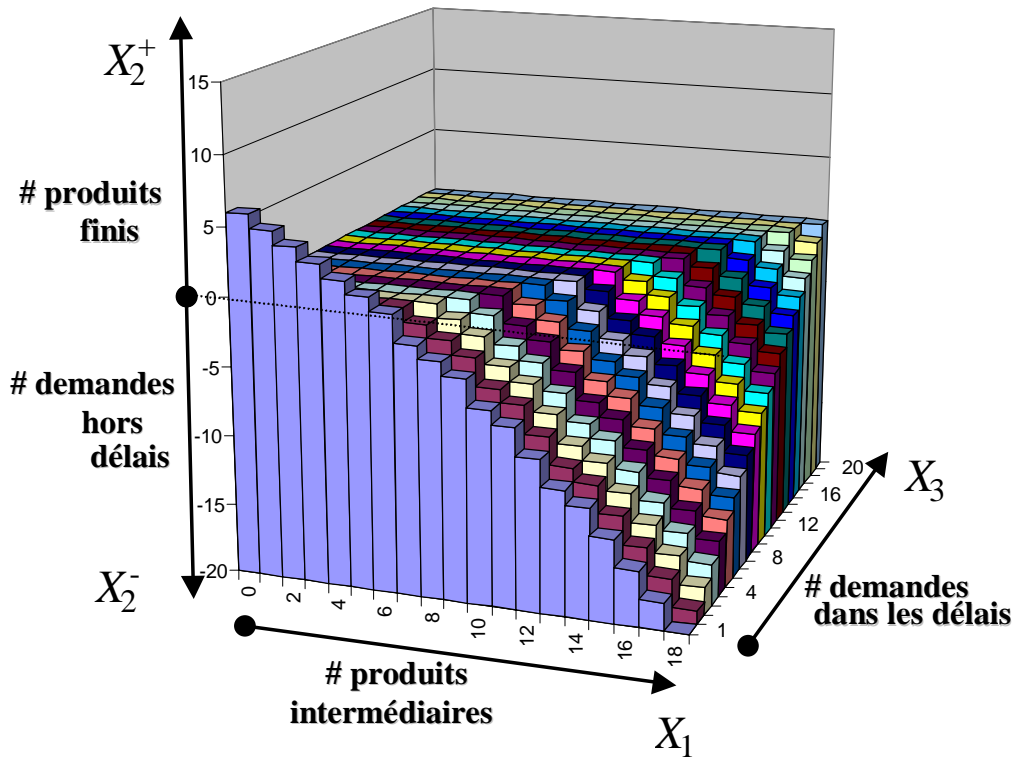


Figure 6.13 : Courbe de commande optimale de la station 1 avec $\mu_1=1.2, \mu_2=1.2, \lambda=1$ et $\lambda_{\text{délai}}=0.1$ c'est-à-dire un délai moyen de 10 unités de temps

On peut voir figure 6.12 et 6.13 deux exemples de courbes de commande optimale pour la station 1, elles sont obtenues par programmation dynamique pour le coût moyen. La composante supplémentaire X_3 multiplie le nombre d'états et augmente d'autant les temps de calcul. Dans le plan $X_3=0$ les courbes de commande des stations 1 et 2 ressemblent à celles du système sans délai. Pour X_3 non nul, les courbes de commande ne dépassent pas l'ordonnée $X_2=0$ car il n'est pas possible d'avoir à la fois des demandes en attente et des produits finis.

L'introduction du délai supprime une partie des coûts liés aux demandes non servies. Ainsi on peut observer une diminution des coûts avec l'augmentation des délais (figure 6.14). Au niveau du plan $X_3=0$, on peut aussi voir (figure 6.12 et 6.13) qu'il s'opère une translation des courbes de commande optimales vers le bas de l'axe X_2 . Ceci correspond à une réduction des stocks de sécurité lorsque le délai augmente.

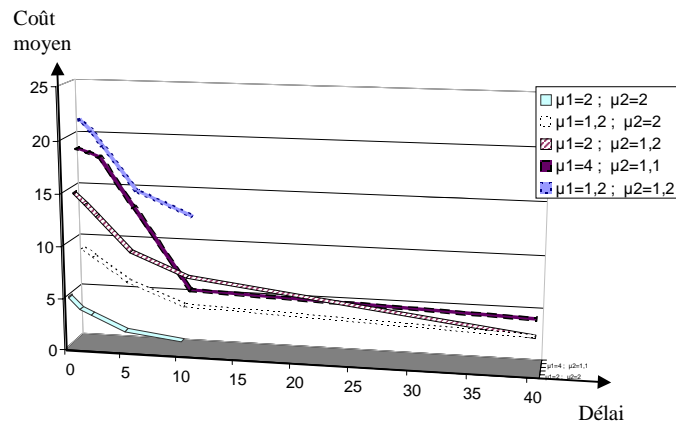


Figure 6.14 : Coût moyen de la politique optimale en fonction du délai moyen

Lorsque X_2 est négatif, pour une valeur fixée de X_1 , on observe que la courbe optimale correspond à une valeur constante de $(X_2 - X_3)$, ceci peut se voir figure 6.12 ou 6.13. Par exemple figure 6.13, la courbe de commande commence en $X_2 = -20$ avec $X_1 = 18$ et $X_3 = 0$, on a donc pour $X_1 = 18$ et $0 \leq X_3 \leq 20$ une courbe de commande optimale qui correspond à $(X_2 - X_3) = -20$. Si cette propriété que nous avons observée sur des résultats numériques était démontrée (nous n'y sommes pas parvenus), alors la mise en œuvre d'une politique optimale ne dépendrait que de la connaissance des inventaires (X_1 et X_2^+) et des arriérés de livraison $(X_2 - X_3)^-$. La valeur $(X_2 - X_3)^-$ représente les demandes non servies indépendamment de l'écoulement du délai. On rappelle que la valeur X_3 ne génère aucun coût alors que X_2^- en génère un selon un taux de valeur b , il n'est donc pas évident que l'une et l'autre de ces valeurs puissent avoir une influence identique sur le pilotage du système de production.

Dans l'article [HAR 1995] il est montré, dans plusieurs cas particuliers, qu'en présence d'un délai la meilleure politique pour lancer un ordre de production consiste à prendre en compte l'arrivée d'une demande sans attendre que cette demande ne devienne hors délai. Ceci va dans le sens de nos observations mais il n'a pas été possible de réutiliser ces résultats pour démontrer ce qui a été observé, car le type de système qu'ils ont étudié est assez différent du notre. Nous montrons qu'une des particularités de leur système, est l'interdiction de livrer avant la fin du délai, comme dans le cas étudié en section 6.5.2, mais il y a d'autres caractéristiques du système qui rendent ces résultats non transposables pour les cas étudiés dans notre travail.

Le calcul du coût des politiques optimales est utilisé dans la section suivante comme valeur de référence pour la comparaison de politiques sous-optimales.

6.5.1.3 Comparaisons

Le calcul des coûts à comparer a été effectué pour plusieurs valeurs du délai et pour plusieurs politiques (le basestock, le « fixed buffer », le kanban et le kanban étendu sont considérées) dont il faut trouver le paramétrage optimal. La politique « fixed buffer » ou

« finite buffer » est étudiée pour un cas sans délai dans [VEA 1994]. Elle consiste à limiter la capacité des stocks de produits finis et de produits intermédiaires (voir section 2.5.4).

On observe souvent une diminution importante du coût moyen de la politique optimale quand le délai augmente, ceci accentue les écarts de coûts en pour-cent par rapport à l'optimal pour le basestock et le kanban. Cependant le kanban généralisé semble rester assez proche des coûts de la politique optimale. Ce phénomène est visible sur la figure 6.15 mais aussi en effectuant une comparaison entre les valeurs statistiques d'un système sans délai (tableau 6.2 de la page 140) et d'un système avec délai moyen de 10 (tableau 6.3).

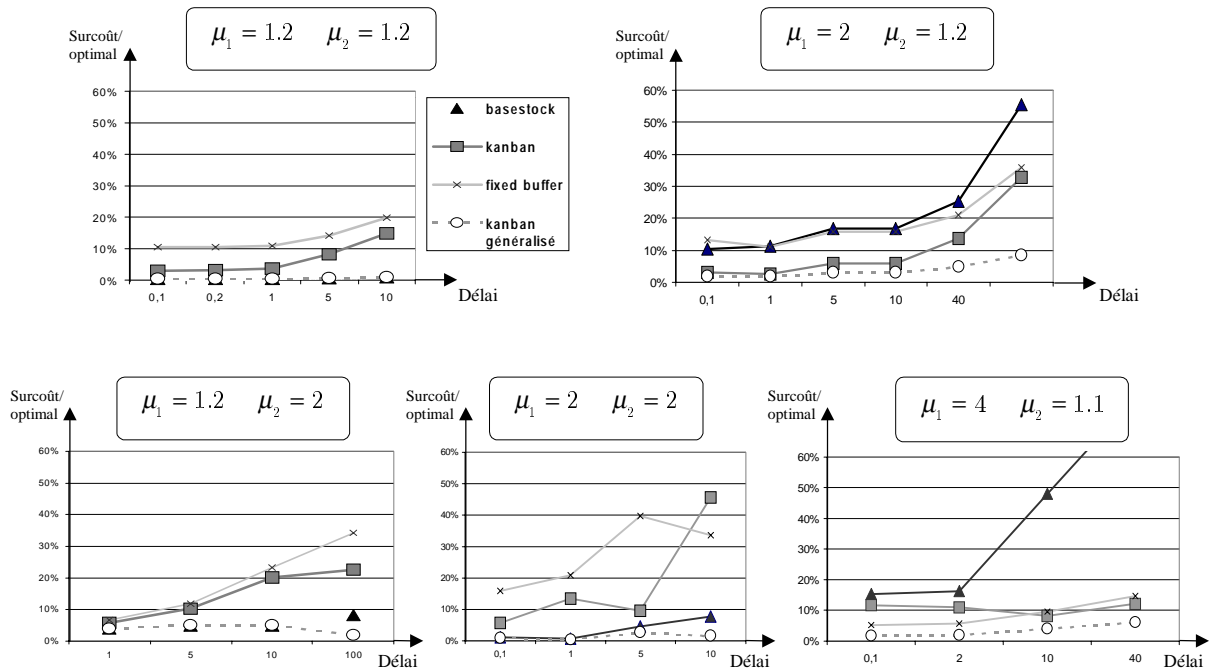


Figure 6.15 : Surcoût par rapport à l'optimal moyen en fonction du délai pour différents systèmes et différentes politiques

Surcoût/optimal:	basestock	kanban	Kanban généralisé
Maximum	48.11%	45.65%	5.06%
Moyen	17.40%	20.49%	3.32%
Minimum	0.87%	8.21%	0.87%

Tableau 6.3: Valeurs maximale, moyenne et minimale du surcoût de différentes politiques par rapport à la politique optimale pour $\lambda_{délai}=0,1$ (délai de 10).

Les politiques sous-optimales utilisées réagissent aux demandes, or dans le cas avec délai deux types de demande existent. Faut-il utiliser les demandes avant délai, après délai ou les deux types de demande ? Nous avons donc fait un choix pour définir les politiques sous-optimales. Puisque les politiques optimales semblent ne dépendre que des inventaires (X_1 et

X_2^+) et des arriérés de livraison $(X_2 - X_3)^-$ (cela a été remarqué dans la section précédente) alors nous avons fait en sorte qu'il en soit de même pour les politiques sous-optimales.

Les courbes de commande des politiques sous-optimales que nous avons définies pour les systèmes avec délais ne dépendent que des valeurs X_1 et $(X_2 - X_3)$. Cela est illustré figure 6.16. Ainsi, chacune des politiques n'utilise que le nombre de demandes non servies $(X_2 - X_3)^-$ sans distinguer celles qui sont avant ou après délai. Les politiques correspondent en fait exactement aux politiques habituelles et seul le calcul du coût est modifié par ce délai.

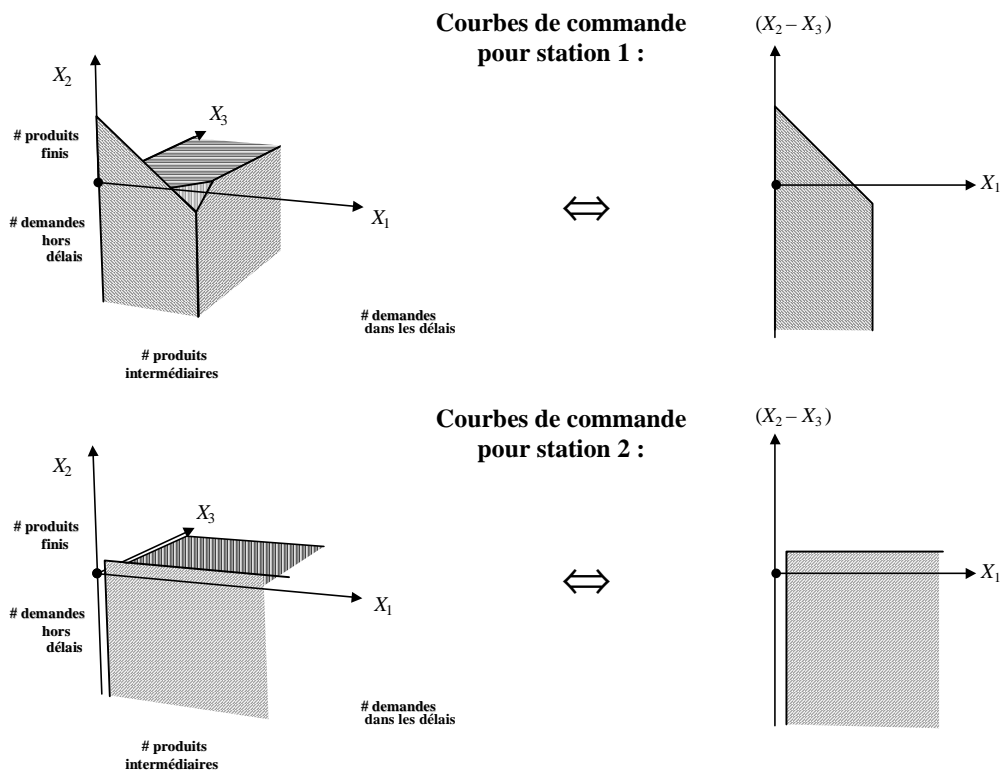


Figure 6.16 : Pour une livraison au plus tôt, courbe de commande d'un kanban généralisé de la station 1 en haut et de la station 2 en bas, avec le système d'axes (X_1, X_2, X_3) à gauche et avec de façon équivalente le système d'axes $(X_1, X_2 - X_3)$ à droite.

6.5.2 Cas du système avec livraison après délai

6.5.2.1 Description

Nous considérons ici le cas où il n'est pas possible de satisfaire une demande tant que le délai qui lui est associé n'est pas écoulé. Le réseau de Petri de la figure 6.17 illustre le fonctionnement d'une ligne de deux étages avec de tels délais. Chaque demande arrive avec un taux λ , puis on a un délai distribué selon une loi exponentielle de taux $\lambda_{\text{délai}}$. Dans la

figure 6.17, il n'y a pas la transition pour une livraison avant délai de la figure 6.11. Il est possible d'avoir à la fois des produits finis et des demandes non servies, donc X_3 peut être non nul lorsque X_2 est positif.

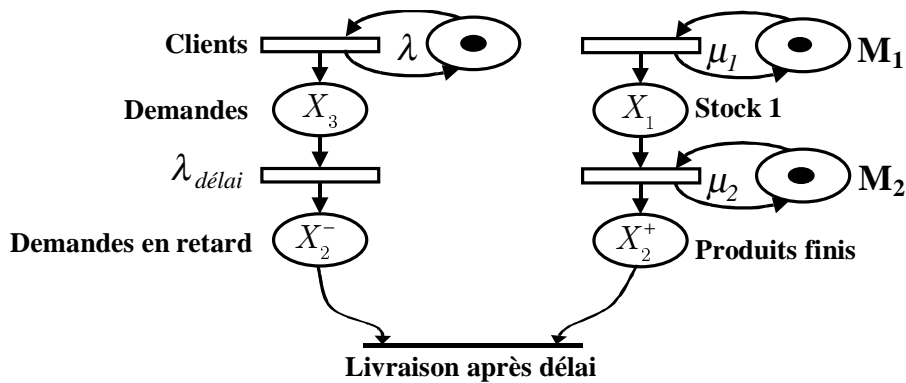


Figure 6.17 : Ligne composée de deux stations M_1 et M_2 imposant un délai de livraison

En utilisant les équations de programmation dynamique (6.8) et (6.18), on peut calculer respectivement le coût pondéré et le coût moyen. Suivant l'état du système ces équations varient en fonction des transitions possibles :

- $d_\lambda = (0, 0, 1)$, $d_{\lambda_{délai}} = (0, -1, -1)$, $d_{\mu_1} = (1, 0, 0)$ et $d_{\mu_2} = (-1, 1, 0)$.
- Si $X_3 > 0$ alors X_3 demandes attendent l'écoulement du délai, chacune peut provoquer une transition de direction $d_{\lambda_{délai}}$. Ainsi le temps passé entre l'écoulement de deux délais suit une loi exponentielle de taux $X_3 \cdot \lambda_{délai}$.

On peut voir figure 6.18 un exemple de courbe de commande optimale obtenue par programmation dynamique. Le profil de la courbe de commande n'est pas tronqué pour X_2 positif comme cela se produit lorsqu'il y a livraison au plus tôt.

On observe, comme lorsqu'il y a livraison au plus tôt avec délai, une diminution des coûts de la politique optimale avec l'augmentation des délais (figure 6.19).

Sur la figure 6.18, on observe que pour $X_1=8$ et $8 < X_3 < 21$, on a $X_2 - X_3 = 12$ (pour $X_3 < 8$ et $X_1=8$ il n'y a pas de valeurs X_2 inférieures à -20 car l'espace d'état est tronqué) ; pour $X_1=7$ on a $X_2 - X_3 = -20$ si $X_3 < 4$, $X_2 - X_3 = -21$ si $3 < X_3 < 15$ et $X_2 - X_3 = -22$ si $14 < X_3 < 21$. Pour une valeurs de X_1 fixée on observe donc que la courbe de commande de la politique optimale suit sur certains intervalles une droite sur laquelle la valeur $(X_2 - X_3)$ est constante. Plus précisément, pour X_1 fixé les points de la courbe sont tels que $(X_2 - X_3)$ est constant pour plusieurs valeurs consécutives de X_3 mais lorsque X_3 croît $(X_2 - X_3)$ décroît parfois d'une unité. On a observé sur des exemples non illustrés ici que plus le délai était important et plus $(X_2 - X_3)$ avait tendance à décroître pour X_1 fixé et X_3 croissant. Cela signifie (surtout quand le délai est important) que la courbe de commande optimale ne dépend pas uniquement de X_1 et $(X_2 - X_3)$ mais plutôt de X_1 , X_2 et X_3 .

Courbe de commande machine amont

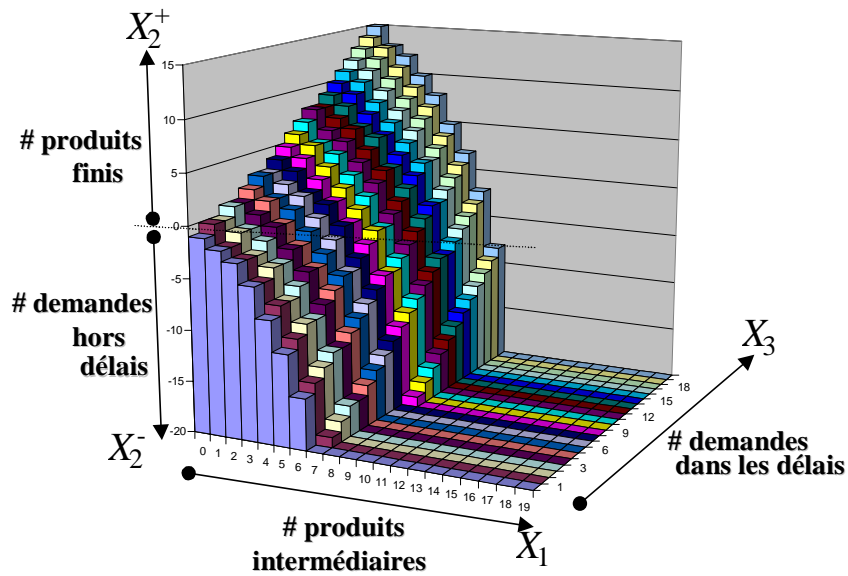


Figure 6.18 : Courbe de commande optimale de la station 1 avec $\mu_1=1,2$ et $\mu_2=1,2$ et un délai moyen de 10 unités de temps

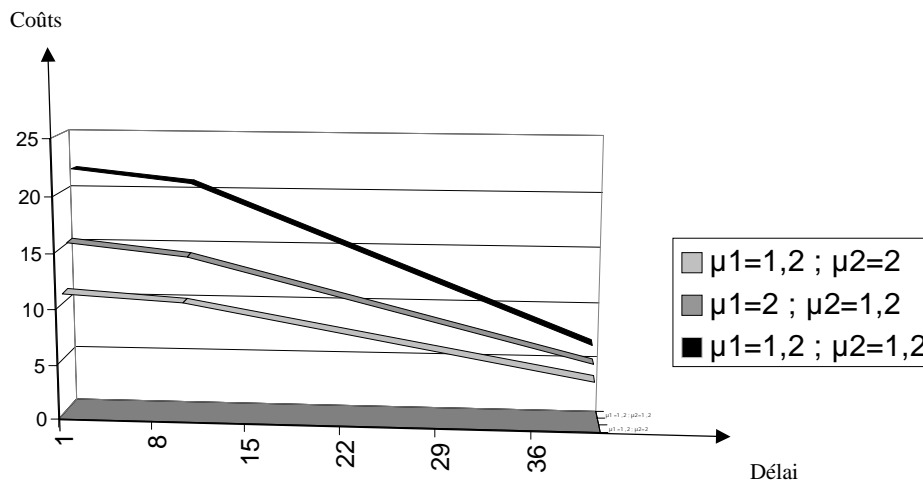


Figure 6.19 : Coût moyen de politiques optimales en fonction du délai moyen

6.5.2.2 Comparaisons

Le calcul du coût des politiques sous-optimales et de la politique optimale a été effectué pour des systèmes avec différents taux de production et différents délais (figure 6.26, page 152). Pour le calcul du coût d'une politique sous-optimale il faut commencer par définir les courbes de commande associées. Comme pour le cas des systèmes avec délai et livraison au plus tôt, nous avons repris les formules des courbes de commandes du cas sans délai en remplaçant X_2 par $(X_2 - X_3)$. Cela signifie que la demande est prise en compte immédiatement par la

politique sans attendre qu'elle devienne hors délai. Les directions de transition $d_{\lambda_{\text{délai}}}$, d_{λ} , d_{μ_1} et d_{μ_2} sont définies dans la section précédente 6.5.2.1, elles autorisent le passage vers des états où X_2 et X_3 sont simultanément strictement positifs (présence simultanée de produits finis et de demandes en attente). Les courbes de commande, pour un système avec livraison après délai, peuvent donc passer par des états où les valeurs X_2 et X_3 sont strictement positives, cela n'est pas possible pour un système avec délai et livraison au plus tôt. Par exemple, avec une politique kanban généralisé on obtient les courbes de commande de la figure 6.20.

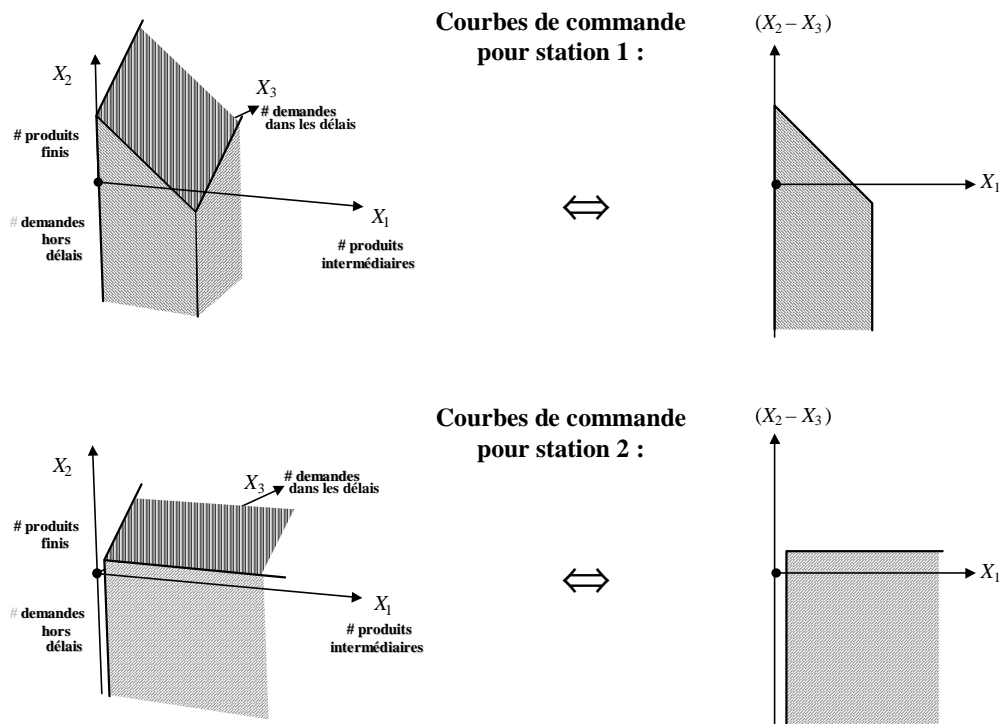


Figure 6.20 : Pour une livraison après délai et prise en compte immédiate des demandes, on a les courbes de commande d'un kanban généralisé dans un repère (X_1, X_2, X_3) à gauche et de façon équivalente dans un repère $(X_1, X_2 - X_3)$ à droite.

Les courbes de commande définies sur les axes X_1 et $(X_2 - X_3)$ correspondent à une prise en compte immédiate des demandes dans un système où la livraison doit s'effectuer après délai. Sur les figures 6.21, 6.22 et 6.23, on peut voir quelles sont les conséquences de cette prise en compte immédiate des demandes sur des politiques kanban, kanban généralisé et basestock modélisées par un réseau de files d'attente. Pour une politique comme le kanban, il faut introduire P_3 qui est un deuxième stock de produits finis. Dans ce stock les produits sont en attente d'une demande dont le délai est en train de s'écouler (figure 6.21). Une politique comme le kanban réagit à la consommation des produits finis. Ici cette consommation s'effectue après l'écoulement du délai associé à la demande. Pour que le kanban réagisse dès l'arrivée de la demande nous avons donc introduit une consommation fictive qui fait passer les produits finis de P_2 à P_3 sans attendre la fin du délai associé à la demande. Avec le

basestock et le kanban généralisé, il n'est pas nécessaire d'introduire un stock supplémentaire P_3 (voir figure 6.22 et 6.23) car ces politiques ne réagissent pas à la consommation des pièces.

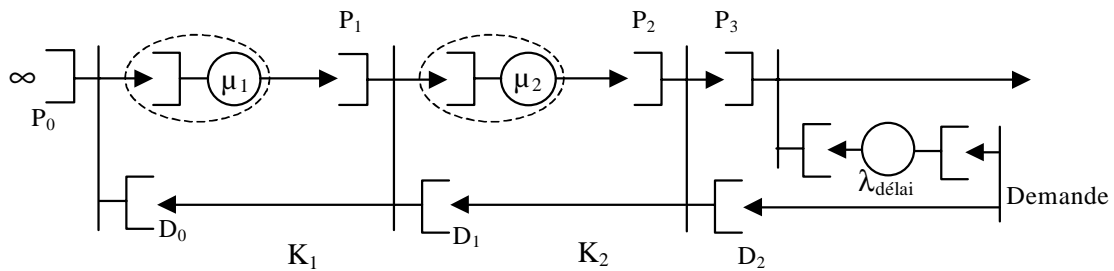


Figure 6.21 : Kanban pour une ligne de deux étages avec livraison après délai et prise en compte immédiate des demandes par la politique.

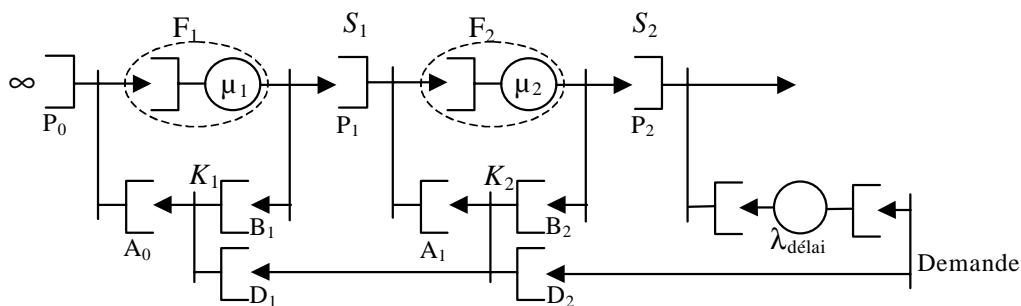


Figure 6.22 : Kanban généralisé pour une ligne de deux étages avec livraison après délai et prise en compte immédiate des demandes par la politique.

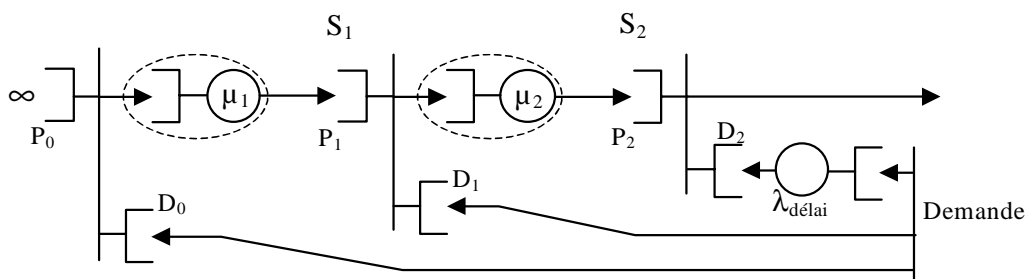


Figure 6.23 : Basestock pour une ligne de deux étages et livraison après délai et prise en compte immédiate des demandes par la politique.

Nous avons choisi pour les politiques sous-optimales une prise en compte immédiate des demandes, alors qu'il peut être possible de choisir une prise en compte après délai des demandes comme cela peut se voir sur la figure 6.24 pour le kanban. Les politiques qui ne réagissent pas aux demandes avant délai X_3 tant qu'elles ne se sont pas transformées en demandes après délai X_2^- , ont une courbe de commande invariante dans le repère (X_1, X_2) quelle que soit la valeur prise par X_3 (voir figure 6.25).

Nous avons vu précédemment que les courbes de commande optimale (surtout quand le délai est important) ne dépendaient pas uniquement de X_1 et $(X_2 - X_3)$. Le choix, pour les

politiques sous-optimales d'une prise en compte immédiate des demandes n'est donc pas toujours approprié et cela particulièrement quand le délai augmente. Cependant ce ne serait pas un meilleur choix que de considérer, comme sur la figure 6.25, que les politiques sous optimales ne sont pas fonctions des demandes avant délai X_3 , vu que les courbes de commande optimales ne sont pas indépendantes de la valeur de X_3 .

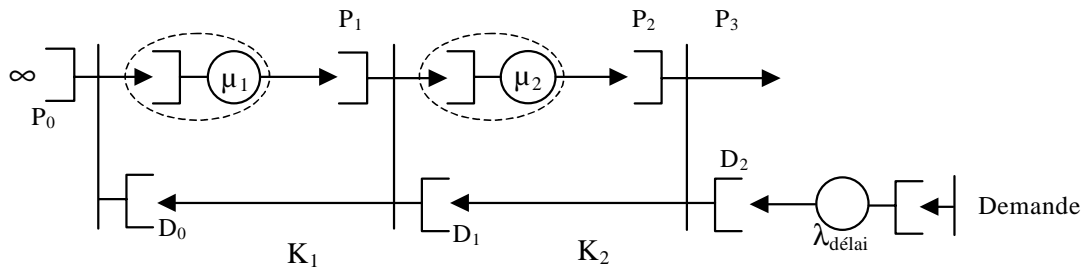


Figure 6.24 : Kanban pour une ligne de deux étages avec livraison après délai et prise en compte après délai des demandes par la politique.

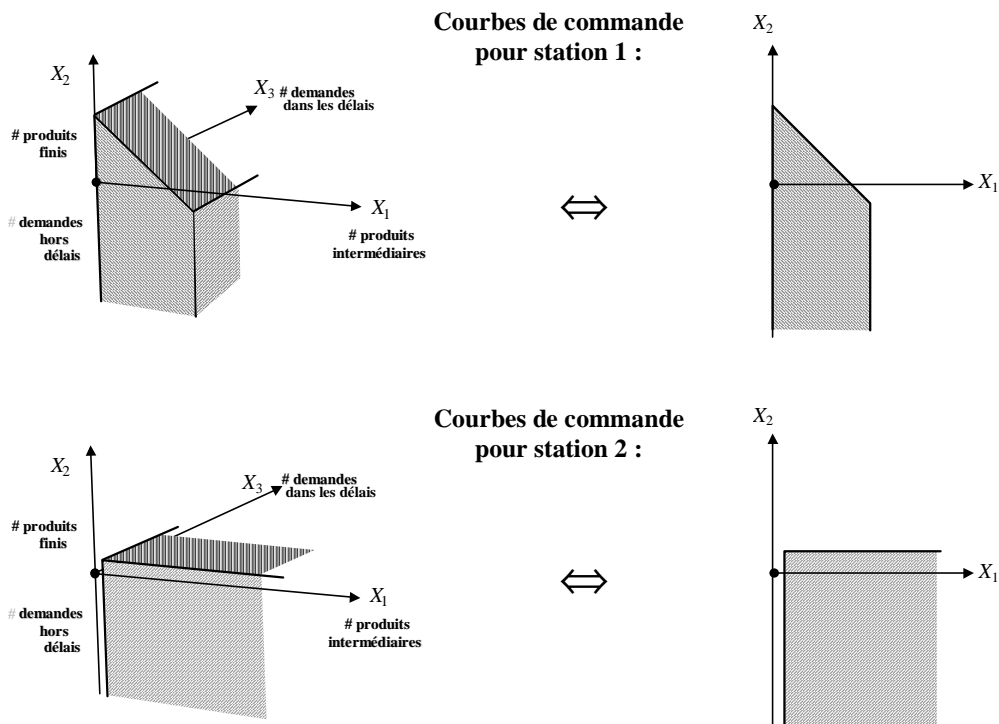


Figure 6.25 : Pour une livraison après délai et prise en compte après délai des demandes, on a les courbes de commande d'un kanban généralisé dans un repère (X_1, X_2, X_3) à gauche et l'équivalence dans un repère (X_1, X_2) à droite.

Sur la figure 6.26, on observe que le kanban généralisé se comporte toujours mieux que le kanban et le basestock. Cependant l'augmentation du délai ne permet pas d'augmenter les performances du kanban généralisé vis-à-vis des autres politiques et de plus le surcoût moyen par rapport à l'optimal croît de façon importante pour les trois politiques sous-optimales

considérées. L'augmentation importante du surcoût par rapport à l'optimal peut s'expliquer par le fait que lorsque le délai est important, il n'est peut être pas idéal d'avoir des politiques sous-optimales qui prennent en compte immédiatement toute demande et qui ne dépendent donc que de X_1 et $(X_2 - X_3)$. On pourrait proposer des politiques sous optimales ne dépendant pas des demandes avant délai X_3 (voir figures 6.25 et 6.24), mais cela n'est intéressant que dans quelques cas où le délai est important.

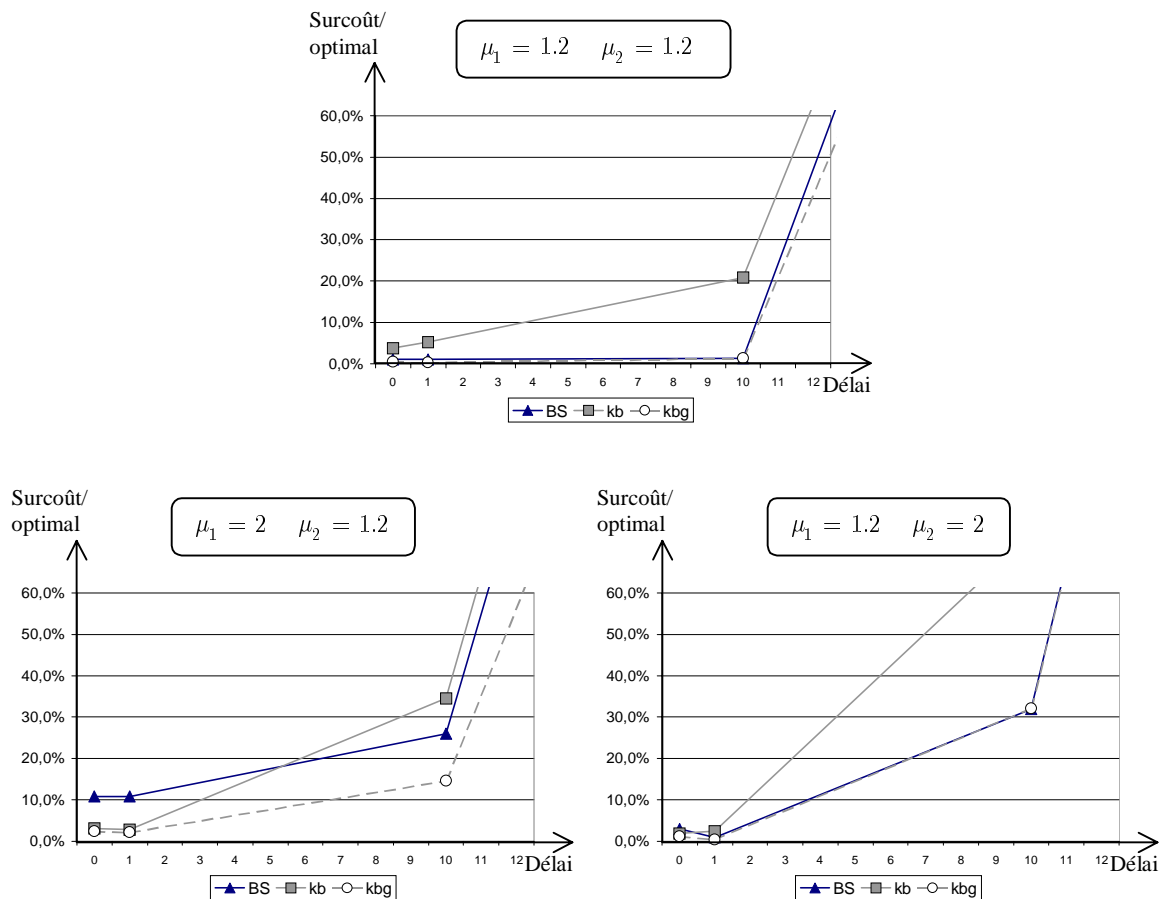


Figure 6.26 : Surcoût moyen par rapport à l'optimal en fonction du délai pour différents systèmes et différentes politiques

6.6 Comparaisons pour un système de deux étages avec production par lots et délai avec livraison au plus tôt

6.6.1 Description

Le système de production considéré ici est composé de deux stations, 1 et 2, chacune comportent un serveur qui traite les pièces selon des lots de taille respectivement Q_1 et Q_2 , le temps de service de ces lots suit une loi exponentielle. La quantité Q_1 est multiple de Q_2 pour

éviter qu'il ne reste dans le stock intermédiaire des pièces ne pouvant être prises dans un lot de taille Q_2 . La demande s'effectue selon des lots de taille 1. Avec de telles hypothèses, X_1 sera toujours multiple de Q_2 et X_2 sera entier naturel quelconque. Pour ce système nous avons introduit d'un délai de livraison, mais seul le cas d'une livraison au plus tôt a été considéré.

6.6.2 Politique optimale

En utilisant les équations de programmation dynamique (6.8) et (6.18) on peut calculer respectivement le coût pondéré et le coût moyen. Suivant l'état du système ces équations varient en fonction des transitions possibles.

- Si $X_2 > 0$ alors $d_\lambda = (0, -1, 0)$, $d_{\lambda_{\text{délai}}} = (0, 0, 0)$, $d_{\mu_1} = (Q_1, 0, 0)$ et $d_{\mu_2} = (-Q_2, Q_2, 0)$.
- Si $-Q_2 \leq X_2 \leq 0$ alors $d_\lambda = (0, 0, 1)$, $d_{\lambda_{\text{délai}}} = (0, -1, -1)$, $d_{\mu_1} = (Q_1, 0, 0)$ et $d_{\mu_2} = (-Q_2, Q_2 - \min(Q_2 + X_2, X_3), -\min(Q_2 + X_2, X_3))$.
- Si $X_2 < -Q_2$ alors $d_\lambda = (0, 0, 1)$, $d_{\lambda_{\text{délai}}} = (0, -1, -1)$, $d_{\mu_1} = (Q_1, 0, 0)$ et $d_{\mu_2} = (-Q_2, Q_2, 0)$.
- Si $X_3 > 0$ alors chacune des X_3 demandes peut devenir une demande ferme et provoquer une transition de direction $d_{\lambda_{\text{délai}}}$. Ainsi la durée des délais suit une loi exponentielle de taux $(X_3 \cdot \lambda_{\text{délai}})$.

Pour le cas $-Q_2 \leq X_2 \leq 0$, c'est-à-dire lorsqu'il y a un petit nombre de demandes hors délais, l'arrivée d'un lot de taille Q_2 peut servir à satisfaire toutes les demandes hors délai et un nombre égal à $\min(Q_2 + X_2, X_3)$ de demandes dans les délais. De plus, si $Q_2 + X_2 > X_3$ alors $(Q_2 + X_2 - X_3)$ pièces de ce lot seront mises dans le stock de produits finis. Pour avoir des résultats comparables à ceux obtenus pour des lots de taille 1, nous avons divisé les taux de service par la taille des lots produits. Nous avons pris $Q_2=3$ et $Q_1=9$. Les demandes arrivent selon un processus poissonien de taux $\lambda=1$, nous avons alors choisi $\bar{\mu}_1 = 1.2/9$ ou $2/9$ et $\bar{\mu}_2 = 1.2/3$ ou $2/3$ pour les taux de service des stations exponentielles. Les courbes de commande optimale obtenues ressemblent à celles des systèmes sans lots avec pour principale différence une commande qui ne s'applique qu'aux états multiples de Q_2 sur l'axe X_1 (figure 6.27).

Courbe de commande machine amont

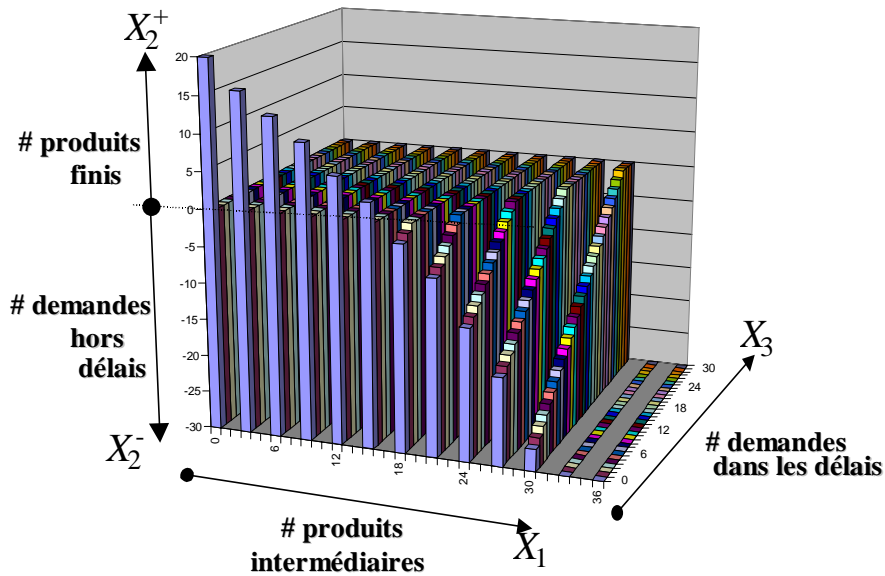


Figure 6.27 : Courbe de commande optimale de la station 1 avec $\mu_1=2/9$ et $\mu_2=1.2/3$ et un délai moyen de 1 unité de temps

On a traité plusieurs exemples et à chaque fois on observe que pour X_2 négatif et X_1 fixé la courbe de commande a une valeur $(X_2 - X_3)$ constante. Les politiques optimales observées semblent donc ne pas faire de différence entre les demandes avant et après délai.

On observe aussi que les courbes de commande optimale ont une ordonnée X_2 qui a tendance à décroître selon des multiples de Q_2 pour X_3 nul (voir tableau 6.4). Ceci peut s'interpréter comme s'il était optimal d'avoir un niveau du stock de produits finis et un nombre de demandes non satisfaites multiple de Q_2 .

Dans un premier temps nous avons optimisé le paramétrage des politiques sous-optimales sans modifier notre heuristique dont le principe consiste à vérifier si localement un paramétrage est optimal. Pour ce type de système produisant par lots, nous avons malheureusement constaté la présence de nombreux minima locaux empêchant le bon fonctionnement de cette heuristique. Cependant pour les politiques kanban et basestock une amélioration de cette heuristique a été trouvée : elle consiste à optimiser le paramétrage en ne prenant au niveau de l'étage 1 que des paramètres multiples de Q_2 , cela permet d'éviter un certain nombre de minima locaux.

Station	μ_1	μ_2	Délai	X_2 pour $X_1=0$	X_2 pour $X_1=3$	X_2 pour $X_1=6$	X_2 pour $X_1=9$	X_2 pour $X_1=12$	X_2 pour $X_1=15$	X_2 pour $X_1=18$	X_2 pour $X_1=21$	X_2 pour $X_1=24$	X_2 pour $X_1=27$	X_2 pour $X_1=30$
1	0.13	0.66	40	7	4	1	-3	-6	-9	-12	-16	-20	-24	-28
2	0.13	0.66	40		-1	0	0	0	0	0	0	0	0	0
1	0.22	0.4	40	4	1	-3	-8	-14	-20	-27				
2	0.22	0.4	40		2	3	3	3	3	3				
1	0.22	0.4	1	20	16	13	10	6	3	-2	-6	-12	-18	-27
2	0.22	0.4	1		13	13	14	14	14	14	14	14	14	14

Tableau 6.4: Ordonnée X_2 des courbes de commande optimales en fonction de l'abscisse X_1 (de 0 à 30) pour X_3 nul.

En comparant les figures 6.28 et 6.14 on observe que la contrainte d'une production par lots est assez coûteuse. Cela est certainement renforcé par le fait qu'avec nos hypothèses, la variabilité du nombre de pièces servies est proportionnelle à la taille des lots servis.

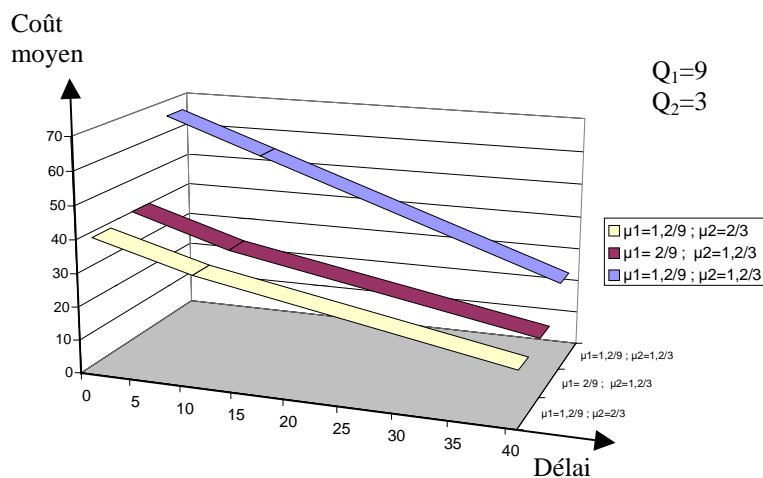


Figure 6.28 : Coût moyen de politiques optimales, avec production par lots, suivant le délai moyen

6.6.3 Comparaisons

Nous avons comparé les surcoûts par rapport à l'optimal pour les politiques kanban et basestock. L'heuristique que nous avons utilisée jusqu'à présent pour calculer le paramétrage optimal des politiques sous optimales n'a pas bien fonctionné avec le kanban généralisé car les coûts obtenus ont été supérieurs à ceux des politiques kanban ou basestock. Pour fonctionner, cette heuristique suppose que le coût moyen est une fonction convexe du paramétrage. Cette

hypothèse est certainement non vérifiée, même approximativement, par le kanban généralisé. Ce phénomène est peut être dû au fait qu'avec cette politique, il y a un nombre de paramètres qui est le double de celui d'un kanban ou d'un basestock.

Les résultats pour le kanban généralisé ne sont donc pas représentés sur la figure 6.29. Les coûts des politiques sous-optimales dans les cas étudiés ne dépassent pas le coût de la politique optimale de plus de 16%.

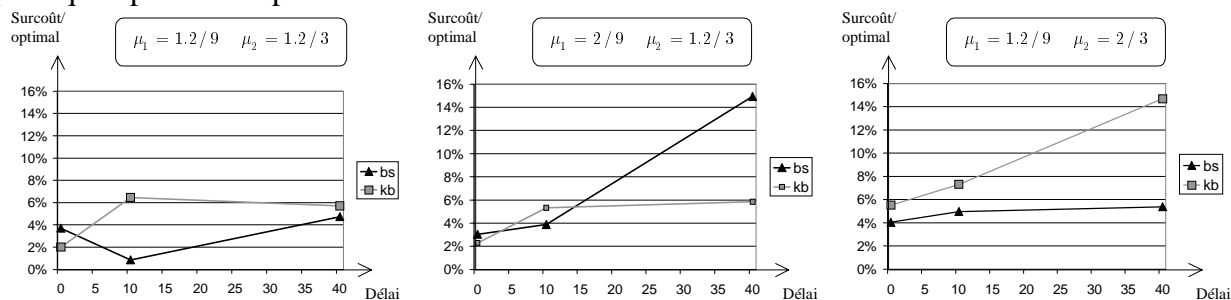


Figure 6.29 : Surcoût/ l'optimal moyen en fonction du délai pour différents systèmes et les politiques kanban(kb) et basestock(bs).

6.7 Système de deux étages avec assemblage

6.7.1 Description

Le système étudié comporte deux stations amont produisant deux produits intermédiaires qui sont assemblés par une station aval (figure 6.30). Ce système comporte trois variables d'état global. Nous n'avons pas introduit de délai car cela nécessiterait l'introduction d'une quatrième variable rendant les calculs numériques trop longs.

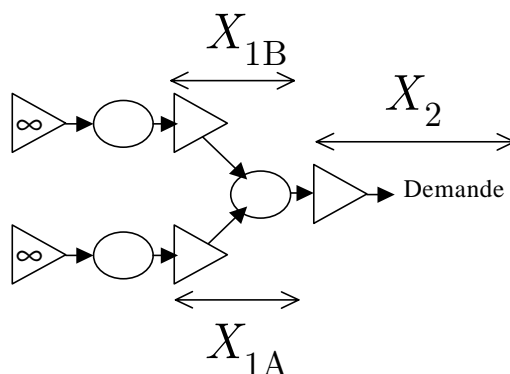


Figure 6.30 : Assemblage de deux produits intermédiaires

Les demandes arrivent selon un processus poissonien de taux λ , les stations sont des serveurs exponentiels. Les deux stations amont ont respectivement un taux de production maximal

$\bar{\mu}_{1A}$ et $\bar{\mu}_{1B}$, celui de la station d'assemblage est $\bar{\mu}_2$. Les taux pour les coûts de possession ont été fixés à 0,5 pour les valeurs X_{1A} et X_{1B} qui représentent les quantités de produits intermédiaires respectivement A et B qui sont en stock ou en cours d'assemblage. Ainsi, chaque produit en cours d'assemblage est pris en compte par X_{1A} et par X_{1B} donc le taux pour le coût de possession des produits assemblés dans la station 2 est de un, c'est-à-dire deux fois 0,5. Pour les produits finis le coût de possession est fixé à $h=2$ et pour les coûts de non satisfaction de la demande on a $b=4$. La formule pour les coûts instantanés est donc :

$$c(X) = 0,5 \cdot X_{1A} + 0,5 \cdot X_{1B} + hX_2^+ + bX_2^-$$

6.7.2 Politique optimale

En utilisant les équations de programmation dynamique (6.8) et (6.18) on peut calculer respectivement le coût pondéré et le coût moyen. Les directions des transitions possibles sont $d_\lambda = (0,0,-1)$, $d_{\mu_{1A}} = (1,0,0)$, $d_{\mu_{1B}} = (0,1,0)$ et $d_{\mu_2} = (-1,-1,1)$.

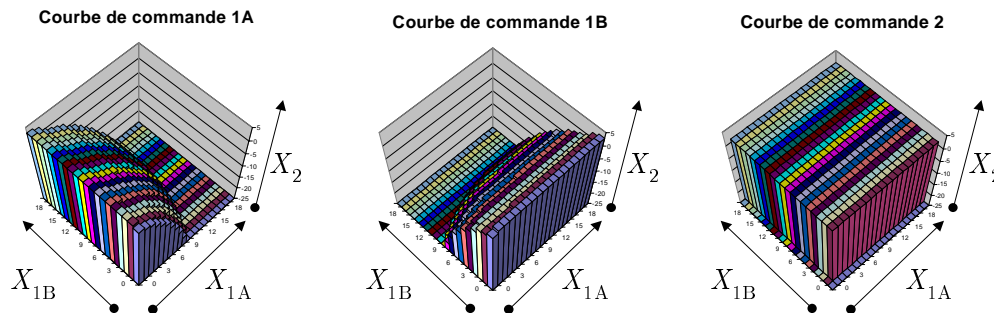


Figure 6.31 : Courbes de commande de chaque station avec $\bar{\mu}_{1A} = \bar{\mu}_{1B} = \bar{\mu}_2 = 2$

Sur l'exemple de la figure 6.31 le profil de la courbe de commande optimale de la station 1A (respectivement 1B) n'est pas indépendant du nombre de produits intermédiaires B (respectivement A) comptabilisés par X_{1B} (respectivement X_{1A}) puisque ce profil varie en fonction de X_{1B} (respectivement X_{1A}). Lorsque X_{1B} est petit, on fabrique moins de produit intermédiaire A que lorsque X_{1B} est grand. Ainsi, il semble qu'il soit nécessaire de maintenir les stocks de produits intermédiaires avant assemblage à des niveaux proches.

La courbe de commande de la station 2 montrée figure 6.31 est identique à celle d'une politique où il faudrait assembler chaque produit A et B disponibles (c'est-à-dire lorsque X_{1A} et X_{1B} sont non nuls) tant que X_2 n'est pas supérieur à 1. Une telle courbe de commande est identique à celle d'une politique sous-optimale où l'objectif de la station 2 est d'atteindre un certain niveau dans le stock de produits finis (un niveau de deux en l'occurrence).

La figure 6.32 montre le coût moyen de plusieurs politiques optimales en fonction des taux de production des stations.

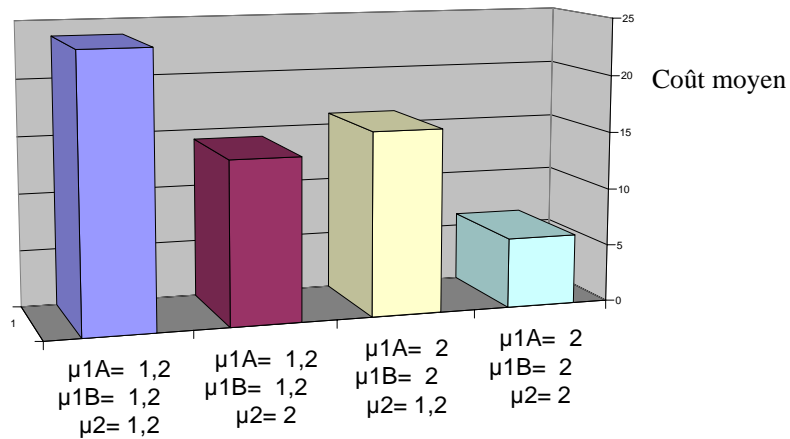


Figure 6.32 : Coût moyen de politiques optimales.

6.7.3 Politique kanban

Les courbes de commande de la politique kanban sont illustrées figure 6.33.

Les zones de marche des stations sont les suivantes :

- Station 1A en marche pour

$$X \in \{ K_{1A} + K_2 - X_{1A} - X_2 > 0, K_{1A} + K_2 - X_{1A} > 0, K_{1A} + X_{1B} - X_{1A} > 0 \}$$

- Station 1B en marche pour

$$X \in \{ K_{1B} + K_2 - X_{1B} - X_2 > 0, K_{1B} + K_2 - X_{1B} > 0, K_{1B} + X_{1A} - X_{1B} > 0 \}$$

- Station 2 en marche pour $X \in \{ K_2 - X_2 > 0, X_{1B} > 0, X_{1A} > 0 \}$.

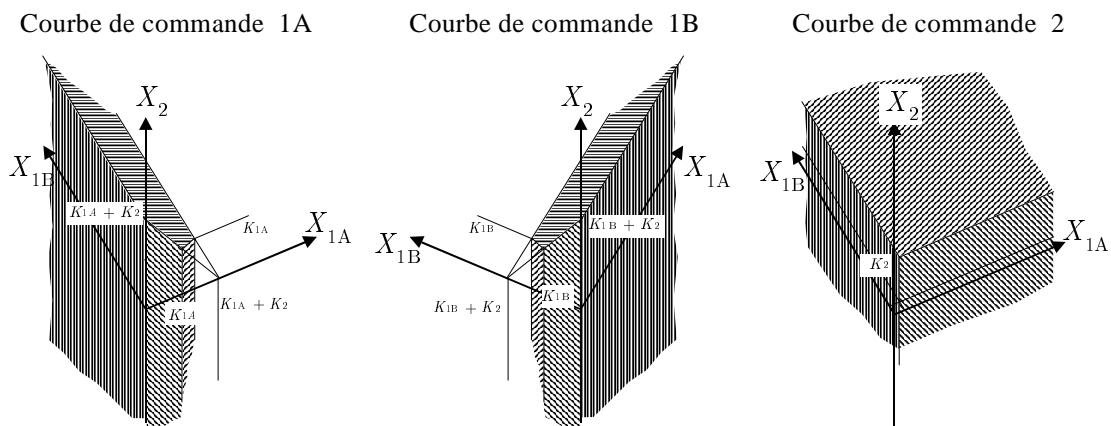


Figure 6.33 : Courbes de commande du kanban pour chaque station

Comme pour la politique optimale, la courbe de commande de la station 1A (respectivement 1B) dépend du nombre de produits intermédiaires B (respectivement A) présents. Plus précisément, la station 1A sera en marche seulement si $K_{1A} + X_{1B} - X_{1A} > 0$, c'est-à-dire

seulement si le nombre de produits intermédiaires a n'est pas supérieur de K_{1A} unités au nombre de produits intermédiaires B.

6.8 Conclusion

Le calcul des politiques optimales par programmation dynamique nous a permis d'avoir une idée de l'intérêt de différentes politiques sous-optimales. Par exemple l'intérêt du kanban généralisé est clairement montré pour des cas avec délai. On a mis en évidence certaines caractéristiques des politiques optimales et on a vérifié si elles sont aussi valables pour les politiques sous-optimales. Par exemple, on a observé avec des calculs numériques qu'en cas de délai avec livraison au plus tôt, une politique optimale réagit aux demandes sans faire la différence entre demandes avant ou après délai. Dans le cas d'une livraison après délai cette observation n'est pas toujours vérifiée, surtout quand le délai augmente. Les politiques sous-optimales proposées ne dépendent que de X_1 et de X_2 ou que de X_1 et $(X_2 - X_3)$. Cela n'est pas vérifié par les politiques optimales lorsqu'il y a livraison après délai avec un délai important car la commande dépend alors à la fois des demandes avant délais X_3 , des demandes après délais X_2^- et des niveaux de stock ou d'encours X_2^+ et X_1 . Ceci explique en partie pourquoi dans certains cas, les politiques sous-optimales étudiées se comportent mal lorsque le délai augmente.

Pour un système de deux étages avec station d'assemblage, on a observé que le contrôle optimal des stations qui précèdent la maille d'assemblage, n'est pas indépendant des différences de niveau entre produits intermédiaires à assembler.

Le cas d'un système produisant par lots a aussi été étudié et des propriétés sur la courbe de commande optimale ont été mises en évidence. Des difficultés ont été rencontrées pour l'optimisation des paramètres du kanban généralisé, en cas de production par lots.

Dans le chapitre suivant, nous avons essayé de calculer les coûts moyens par une autre technique que la programmation dynamique. Nous avons surtout essayé d'obtenir la plus grande précision possible. Nous nous sommes limités à l'étude d'une ligne de deux mailles.

Chapitre 7

Quelques tentatives de développement d'une méthode de calcul des coûts moyens commune à toutes les politiques.

Dans le chapitre précédent nous avons effectué des calculs du coût moyen pour différentes politiques, ces calculs sont réalisés sur un espace d'état tronqué il ne peuvent donc pas être complètement exacts. L'objectif de ce chapitre est de calculer le coût moyen des politiques avec un maximum de précision. Pour obtenir le coût moyen, notre démarche consiste à calculer la probabilité stationnaire de chaque état. Il faut donc résoudre une chaîne de Markov infinie qui représente tous les changements d'état possibles d'un système de production donné. Pour limiter la complexité de cette chaîne de Markov, nous nous sommes limités à l'étude d'un système de production constitué de deux mailles.

Cette chaîne de Markov s'obtient à partir des $F_i(X)$ qui nous indiquent la marche et l'arrêt des stations. Sa structure matricielle est répétitive et infinie. Pour la résoudre nous avons utilisé la méthode de la matrice géométrique de Neuts [NEU 1981]. Cette méthode est fréquemment utilisée dans la résolution de chaînes de Markov des processus de quasi naissance et mort (voir des exemples d'utilisation dans [DUR 1997] et [BUZ 1993]).

Deux méthodes de calcul sont proposées, elles sont exactes en théorie mais en pratique lors des calculs numériques la précision des résultats obtenus peut varier. Lorsque la valeur exacte du coût est un nombre rationnel il est possible avec une des deux méthodes d'obtenir ce coût sous la forme d'un quotient de deux nombres entiers. Si ce coût n'est pas un nombre rationnel, les deux méthodes proposées ne peuvent alors donner un résultat que sous la forme d'une valeur numérique qui comporte de petites erreurs de précision dues aux calculs numériques. Le calcul de la valeur rationnelle exacte du coût n'a été possible que pour des politiques de gestion de production où $X_1 \leq 2$, c'est-à-dire pour des politiques limitant à deux le nombre de produits présents dans le stock intermédiaire et dans l'encours de la station 2. Même lorsque le résultat n'est pas donné sous la forme d'un quotient, il est nécessaire, pour les deux méthodes de calcul proposées, d'avoir une politique qui limite le nombre de produits présents

dans le stock intermédiaire et dans l'encours de la station 2. Cela signifie qu'il est nécessaire d'avoir X_1 borné.

Dans un premier temps nous décrirons, pour un système de production ayant deux stations exponentielles, la chaîne de Markov de différentes politiques. Des exemples simples et d'autres plus généraux seront étudiés avec les deux méthodes de calcul proposées.

7.1 Utilisation de la méthode de la matrice géométrique de Neuts

Les systèmes étudiés comportent deux mailles. Dans chaque maille i une station exponentielle de taux de service μ_i est à l'arrêt ou en marche suivant que $F_i(X)$ est nul ou non. La demande arrive selon un processus poissonien de taux λ . Le nombre d'états globaux X est infini car il n'y a pas de limite à l'accumulation des demandes non servies (X_2 peut tendre vers moins l'infini). Les états et les transitions possibles dépendent de $F_1(X)$ et $F_2(X)$ qui définissent la politique de gestion de production.

Nous n'étudierons que les politiques où le nombre X_1 de produits dans le stock intermédiaire et dans l'encours de la station 2, est borné par une valeur maximale notée X_1^{\max} . Cela permet d'avoir une structure répétitive de la chaîne de Markov. On supposera aussi que la courbe de commande de la station 2 consiste uniquement à produire tant qu'un niveau noté X_2^{\max} n'est pas atteint dans le stock de produits finis. On a donc $F_2(X) = \min(X_1, X_2^{\max} - X_2)$. En X_2^{\max} la courbe de commande de la station 1 a une certaine abscisse X_1 qui va en augmentant lorsque X_2 décroît ; elle finit par atteindre la valeur X_1^{\max} . A partir de là, la courbe de commande de la station 1 est verticale, elle autorise alors à produire au plus X_1^{\max} produits pour le stock intermédiaire et l'encours de la station 2.

La politique basestock n'est pas abordée car, à cause d'un encours non borné de la station 2 (c'est-à-dire X_1 non borné), la structure de la chaîne de Markov n'est pas répétitive. Cela rend alors inapplicables les techniques de calcul que nous avons trouvées.

Pour effectuer ces calculs, il nous faut définir un générateur infinitésimal. Pour cela il est nécessaire de fixer la numérotation des états. Différentes numérotations sont nécessaires pour chacune des deux méthodes de résolution de la chaîne de Markov que nous avons proposées. La numérotation des états exposée ci-dessous dans la section 7.1.1 permet d'obtenir la chaîne de Markov qui sera résolue en utilisant la méthode de la matrice géométrique de Neuts.

7.1.1 Numérotation des états

Cette méthode de numérotation des états se fait de gauche à droite et de haut en bas. La figure 7.1 illustre une telle numérotation des états globaux atteignables pour une politique

quelconque où $X_1^{\max} = 5$. Sur cette figure on peut voir entre les ordonnées 4 et 5 un trait horizontal qui représente la courbe de commande de la station 2, on a donc $X_2^{\max} = 5$. Les cases grisées représentent les états où les deux stations sont en marche. Pour l'état numéro 1 la station 1 marche tandis que la station 2 est à l'arrêt. Pour l'état numéro 2 les deux stations sont à l'arrêt. Pour l'état numéro 5, la station 2 est en marche et la station 1 est à l'arrêt. On peut voir le détail des transitions possibles sur la partie droite de la figure 7.1.

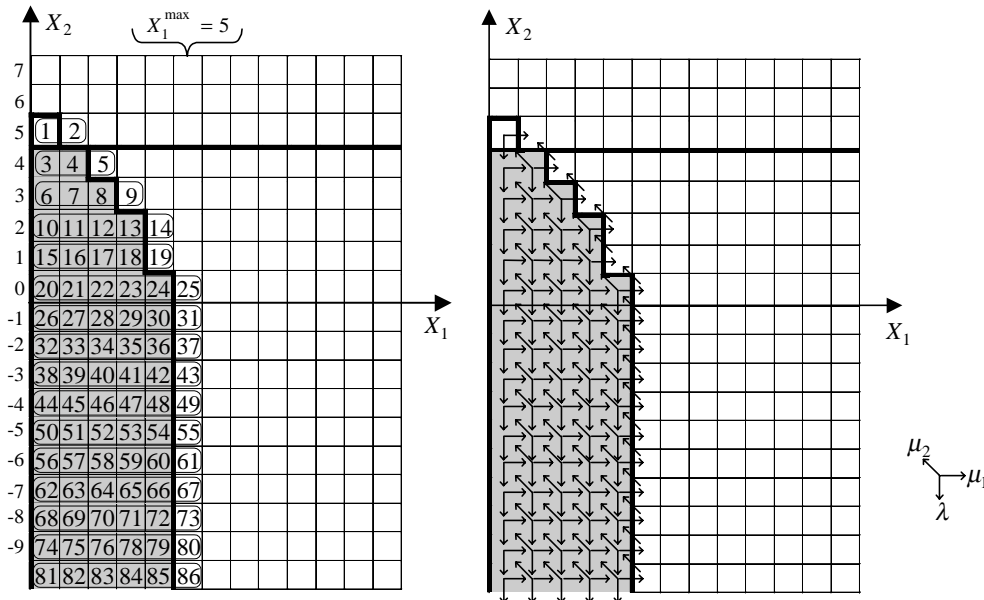


Figure 7.1 : Un exemple de transition et de numérotation des états d'une politique

Un cas particulier de ce genre de politique est le kanban généralisé. Considérons l'exemple de la figure 7.2 qui illustre en parallèle la courbe de commande avec numérotation des états ergodiques, les transitions possibles et le générateur infinitésimal associé à une politique kanban généralisé dont le paramétrage est $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$ (K_1 étant quelconque). Pour cet exemple, on a $X_2^{\max} = 5$ et $X_1^{\max} = 1$.

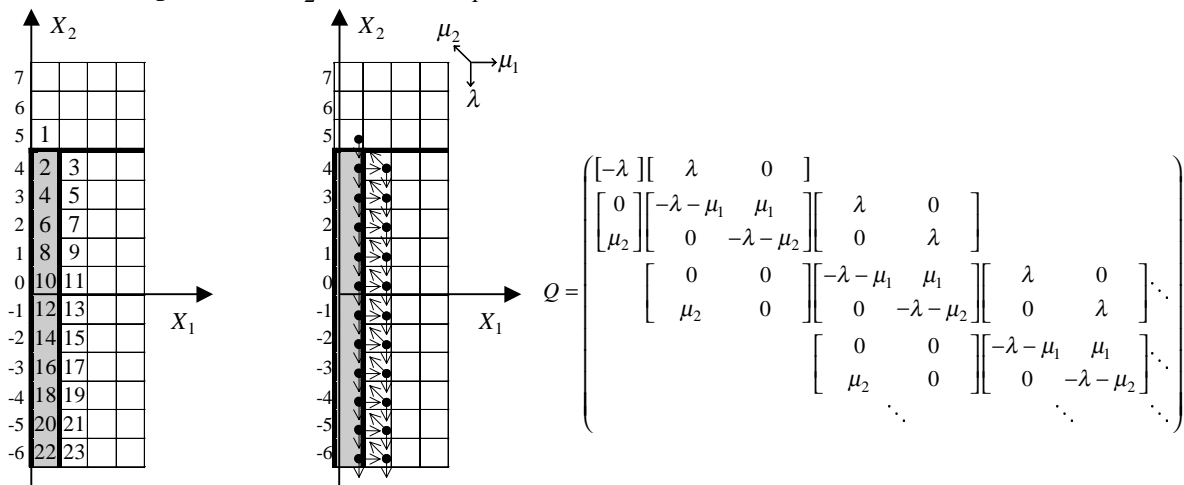


Figure 7.2 : Courbes de commande et générateur infinitésimal d'une politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$.

Le cas $X_1^{\max} = 2$ peut être illustré par une politique kanban généralisé dont le paramétrage est $K_2 = 1$, $S_1 = 1$ et $S_2 = X_2^{\max}$ (K_1 étant quelconque). Dans la figure 7.3 les courbes de commande, la numérotation des états et le générateur infinitésimal d'un kanban généralisé sont représentés pour le paramétrage $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$.

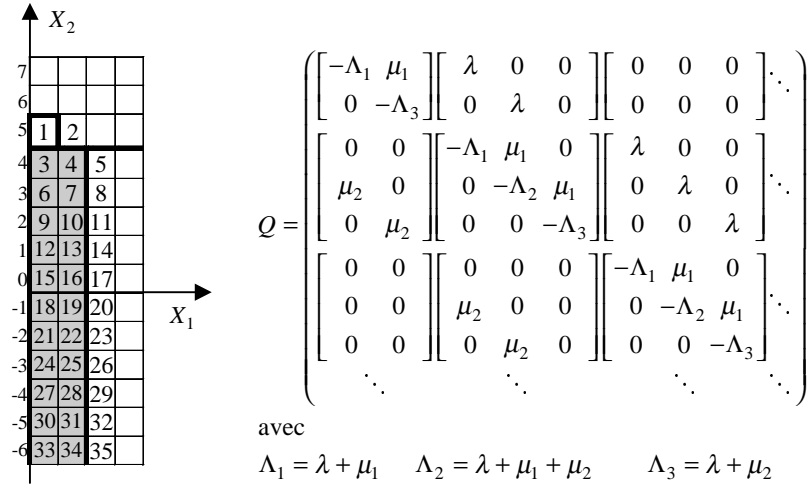


Figure 7.3 : Courbes de commande et générateur infinitésimal du kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$.

On peut généraliser la structure des générateurs infinitésimaux obtenus figures 7.2 et 7.3, à des politiques quelconques où la courbe de commande de la station 2 est horizontale et la courbe de commande de la station 1 devient verticale à partir d'une certaine valeur de X_2 (voir figure 7.1). On obtient alors un générateur infinitésimal dont l'aspect général est le suivant :

$$Q = \begin{pmatrix} B_0 & A_0 & 0 & 0 & 0 & 0 & \ddots \\ C_0 & B_1 & A_1 & 0 & 0 & 0 & \ddots \\ 0 & C_1 & \ddots & \ddots & 0 & 0 & \ddots \\ 0 & 0 & \ddots & B_i & A_i & 0 & \ddots \\ 0 & 0 & 0 & C_i & B_i & A_i & \ddots \\ 0 & 0 & 0 & 0 & C_i & B_i & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \quad (7.1)$$

Les composantes $A_0, A_1, \dots, A_i, B_0, B_1, \dots, B_i, C_0, C_1, \dots$ et C_i sont des sous-matrices. Pour $j < i$ les sous-matrices A_j, B_j et C_j ont respectivement n, n et m lignes et m, n et n colonnes. Le nombre n (respectivement m) correspond au nombre d'états dont l'ordonnée X_2 est $(X_2^{\max} - j)$ (respectivement $(X_2^{\max} - j - 1)$). Si $n = m$ alors les sous-matrices A_j, B_j et C_j ont respectivement des composantes identiques à celle de A_i, B_i et C_i , mais elles ont cependant un nombre de lignes et de colonnes inférieur. Les sous-matrices A_i, B_i et C_i ont X_1^{\max} lignes et X_1^{\max} colonnes, elles sont définies par :

$$C_i = \begin{pmatrix} 0 & & & & & \\ \mu_2 & 0 & & & & \\ & \mu_2 & \ddots & & & \\ & & \ddots & 0 & & \\ & & & \ddots & 0 & \\ & & & & \mu_2 & 0 \end{pmatrix}, B_i = \begin{pmatrix} -\Lambda_1 & \mu_1 & & & & \\ & -\Lambda_2 & \ddots & & & \\ & & \ddots & \mu_1 & & \\ & & & \ddots & \mu_1 & \\ & & & & -\Lambda_2 & \mu_1 \\ & & & & & -\Lambda_3 \end{pmatrix} \text{ et } A_i = \begin{pmatrix} \lambda & & & & & \\ & \lambda & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \lambda & \\ & & & & & \lambda \end{pmatrix}$$

avec $\Lambda_1 = \mu_1 + \lambda$, $\Lambda_2 = \mu_1 + \mu_2 + \lambda$ et $\Lambda_3 = \mu_2 + \lambda$.

Chaque sous-matrice représente des transitions entre sous-ensembles d'états. Ces sous-ensembles sont constitués d'états ayant une même ordonnée X_2 dans l'espace d'état. Ces sous-ensembles d'états sont représentés figure 7.1 par des ovals. Le rang i utilisé dans la matrice (7.1) est égal au nombre d'ordonnées X_2 où il y a moins de $(X_1^{\max} + 1)$ états, c'est-à-dire le nombre de lignes parmi les états ergodiques où la courbe de commande n'est pas encore devenue verticale. Sur la figure 7.1, pour X_2 allant de 5 à 1 il y a moins de 6 états par ligne donc $i = 5$. Sur la figure 7.2, il y a moins de 2 états par ligne que pour $X_2=5$, donc $i = 1$ et sur la figure 7.3, il y a moins de 3 états par ligne que pour $X_2=5$, donc là encore $i = 1$.

En supposant $F_2(X) = \min(X_1, X_2^{\max} - X_2)$, c'est-à-dire que la station 2 marche tant qu'un niveau de produits fini X_2^{\max} n'est pas atteint et en supposant que $F_1(X)$ respecte les propriétés caractérisant les courbes de commande optimale (D-monotonie) alors on a $n = m - 1$ si $n \neq m$. Pour $n = m - 1$ la structure de B_j est la même que celle de B_i , celle de A_j est identique à A_i auquel il est retiré la dernière ligne et celle de C_j est identique à C_i auquel il est retiré la première ligne. La sous-matrice B_0 diffère de B_i par sa diagonale, sa structure est la suivante :

$$B_0 = \begin{pmatrix} -\Lambda_1 & \mu_1 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \mu_1 & & \\ & & & -\Lambda_1 & \mu_1 & \\ & & & & & -\lambda \end{pmatrix}$$

Les générateurs infinitésimaux exposés ci-dessus vont nous permettre de calculer, avec la méthode de la matrice géométrique de Neuts [NEU 1981], les probabilités stationnaires moyennes de chaque état et le coût moyen.

7.1.2 Méthode de calcul

La méthode de Neuts [NEU 1981], ou de la matrice géométrique, permet de trouver les probabilités d'un processus de quasi naissance et mort dont le générateur infinitésimal est le suivant :

$$Q = \begin{pmatrix} G_0 & H_0 & 0 & 0 & \ddots \\ G_1 & H_1 & H_0 & 0 & \ddots \\ 0 & H_2 & H_1 & H_0 & \ddots \\ 0 & 0 & H_2 & H_1 & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \quad (7.2)$$

avec G_0, G_1, H_0, H_1 et H_2 des sous-matrices carrées de dimension identiques. Pour résoudre la chaîne de Markov associée à ce problème le vecteur de probabilité P qui est de dimension infinie, est découpé en sous-vecteurs P_i pour i allant de 0 à l'infini. La taille des vecteurs P_i est la même que celle de G_0, H_0, H_1 et H_2 .

On ne peut trouver des probabilités stationnaires solutions que si la propriété suivante (voir [NEU 1981] page 83) est vérifiée :

$${}^t\pi H_2 e > {}^t\pi H_1 e \quad (7.3)$$

avec π le vecteur probabilité solution pour le générateur $(H_2 + H_1 + H_0)$, ${}^t\pi$ le vecteur transposé de π et e un vecteur dont chacune des composantes vaut un. Ceci permet de vérifier si à partir d'un certain rang, la probabilité mesurée par $P_i e$ est bien décroissante.

Les probabilités stationnaires associées aux sous-vecteurs de P sont alors données par :

$$P_i = P_0 R^i \quad \text{pour } i \geq 0$$

avec R solution de :

$$R^2 H_2 + R H_1 + H_0 = 0$$

et P_0 se calculant par le système d'équations suivant :

$$\begin{cases} {}^t P_0 (G_0 + R G_1) = 0 \\ {}^t P_0 (Id - R)^{-1} e = 1 \end{cases}$$

L'inéquation (7.3) est vérifiée si et seulement si les valeurs propres de R sont strictement à l'intérieur du disque unité. Donc $(Id - R)^{-1}$ existe et quand i tend vers l'infini R^i tend vers zéro, si et seulement si l'inéquation (7.3) est vérifiée. Ainsi on peut montrer que le vecteur $P_i = P_0 R^i$ tend vers zéro. Si celui-ci ne tend pas vers zéro, il n'est pas un sous-vecteur de probabilité.

La méthode de Neuts ne peut pas être utilisée directement pour trouver les probabilités associées au générateur infinitésimal (7.1) car la forme de la matrice n'est répétitive qu'à partir d'un certain rang i . Il est donc nécessaire d'adapter cette méthode de la sorte :

La taille des sous-vecteurs P_j doit être la même que celle de B_j pour $j < i$ et la même que B_i sinon. La solution est alors donnée par :

$$P_j = P_i R^{j-i} \quad \text{pour } j \geq i$$

avec R solution de :

$$R^2 A_i + R B_i + C_i = 0$$

Les sous-vecteurs P_j , pour j allant de 0 à i , sont donnés par le système d'équations suivant :

$$\begin{cases} {}^tP_0B_0 + {}^tP_1C_0 = 0 \\ {}^tP_{j-1}A_{j-1} + {}^tP_jB_j + {}^tP_{j+1}C_j = 0 \text{ pour } j \text{ allant de } 1 \text{ à } i-1 \\ {}^tP_{j-1}A_{i-1} + {}^tP_i(B_i + RC_i) = 0 \\ \sum_{j=0}^{i-1} {}^tP_j e + {}^tP_i (Id - R)^{-1} e = 1 \end{cases}$$

Les trois premières lignes du système sont des équations d'équilibre et la dernière est une équation de normalisation. Pour qu'une solution existe on pourra vérifier la propriété suivante similaire à (7.3) :

$${}^t\pi C_i e > {}^t\pi A_i e \quad (7.4)$$

avec π le vecteur probabilité solution pour le générateur $(A_i + B_i + C_i)$.

Pour calculer R on utilise la suite R_j . Cette suite a pour limite R , elle est strictement croissante et définie par :

$$R_0 = 0, \quad R_{j+1} = (R_j^2 A_i + C_i)(-B_i)^{-1} \quad (7.5)$$

Une valeur non nulle pour R_0 peut causer des problèmes de convergence. Nous effectuons ci-dessous les calculs numériques pour la méthode que nous venons d'exposer.

7.1.3 Résultats numériques

Nous traitons ci-dessous les cas exposés sur les figures 7.2 et 7.3. Sur chacun des exemples il faut choisir des valeurs pour les taux de service des stations.

7.1.3.1 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$.

Pour l'exemple de la figure 7.2 on doit vérifier (7.4) ce qui après simplification équivaut à vérifier :

$$\mu_1 \mu_2 - \mu_1 \lambda - \mu_2 \lambda > 0.$$

Par exemple prendre $\lambda = 1$, $\mu_1 = 2$ et $\mu_2 = 2$ ne convient pas car alors le calcul de R^i ne tend pas vers zéro lorsque i tend vers l'infini. Nous prendrons pour la suite des calculs $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$. Pour commencer il faut calculer R . Nous avons approché la valeur de cette matrice avec R_{100} qui est calculé par une centaine d'itérations, sa valeur est la suivante

$$\begin{pmatrix} 0,4999987648 & 0,4999987648 \\ 0,1666659535 & 0,4999992868 \end{pmatrix}$$

On peut alors calculer le vecteur P :

$$P_0 = [0,1666682177], \quad P_1 = \begin{bmatrix} 0,08333394650 \\ 0,08333394650 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0,05555580197 \\ 0,08333378414 \end{bmatrix} \text{ et } P_i = R^{i-1} P_1$$

Beaucoup de chiffres après la virgule sont donnés pour qu'il soit possible de se faire une idée de la précision des calculs. La matrice R a des composantes dont les valeurs sont proches

de 1/2 ou de 1/6 et P_0 vaut presque 1/6. Sur la figure 7.4 les probabilités sont représentées à l'emplacement de chaque état.

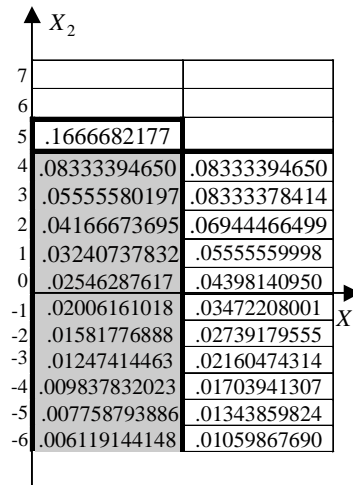


Figure 7.4 : Probabilité de chaque état en fonction des coordonnées pour le kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$. Résultats obtenus par la méthode de la matrice géométrique avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$.

7.1.3.2 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$.

Pour l'exemple de la figure 7.3 il est aussi nécessaire de vérifier (7.4) ; après simplification on trouve :

$$\mu_2^2\mu_1 + \mu_1^2\mu_2 - \mu_2^2\lambda - \mu_2\mu_1\lambda - \mu_1^2\lambda > 0.$$

Par exemple les taux $\lambda = 1$, $\mu_1 = 3/2$ et $\mu_2 = 3/2$ ne conviennent pas car on obtient zéro pour le terme de gauche. Pour traiter ce cas particulier de kanban généralisé on prendra les mêmes valeurs que précédemment c'est-à-dire $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$. On trouve alors, après une centaine d'itérations, la valeur approchée suivante pour R :

$$\begin{pmatrix} .3333333333 & .2499999999 & .2499999999 \\ .04761904763 & .2500000000 & .2500000000 \\ .03174603175 & .08333333335 & .4166666667 \end{pmatrix}$$

On peut alors calculer une partie du vecteur P , les résultats numériques sont montrés figure 7.5.

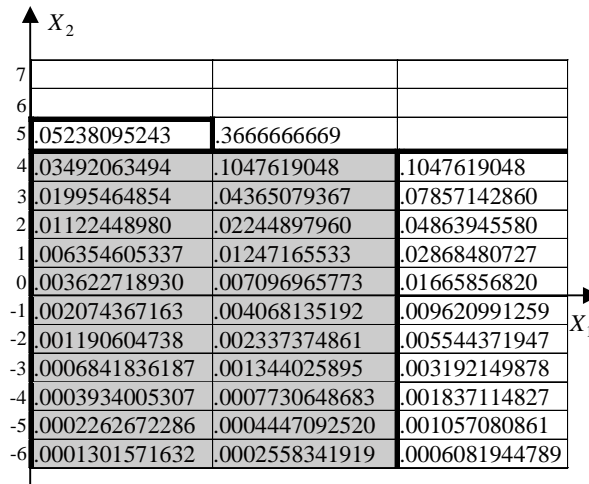


Figure 7.5 : Probabilité de chaque état en fonction des coordonnées pour le kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$. Résultats obtenus par la méthode de la matrice géométrique avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$.

7.1.3.3 Politique kanban généralisé dont les paramètres sont $K_2 = 2$, $S_1 = 1$ et $S_2 = 5$.

La méthode de la matrice géométrique s'applique sans difficulté particulière. Après avoir approximé R par R_{100} (formule (7.5)) on obtient les probabilités illustrées sur la figure 7.6 suivante :

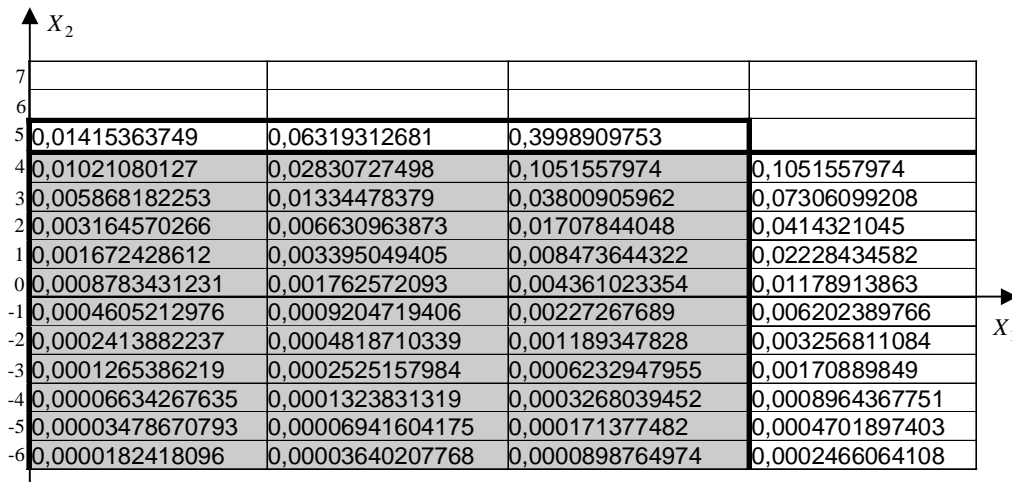


Figure 7.6 : Probabilité de chaque état en fonction des coordonnées pour le kanban généralisé avec $K_2 = 2$, $S_1 = 1$ et $S_2 = 5$. Résultats obtenus par la méthode de la matrice géométrique avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$.

Remarque :

Pour ces trois politiques, il existe un état qui correspond à un nombre de pièces maximal au niveau de chaque échelon c'est-à-dire où X_2 et $X_1 + X_2$ sont maximaux. Cet état est ce que l'on appelle le « hedging point » [KAR 2000], on remarque ici qu'il est à chaque fois l'état le plus probable. Sur la figure 7.6 il s'agit du point d'ordonnée 5 et d'abscisse 2.

7.1.4 Calcul du coût moyen et de quantités moyennes

Pour le calcul du coût moyen on multiplie le taux de coût de chaque état par sa probabilité que l'on peut voir sur les figures 7.4, 7.5 et 7.6. Les coûts instantanés de chaque état sont donnés par $c(X) = X_1 + hX_2^+ + bX_2^-$.

7.1.4.1 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$.

Le coût moyen peut se calculer à partir des valeurs moyennes de X_2^+ , X_2^- et X_1 . Pour calculer ces valeurs moyennes il faut utiliser les sous-vecteurs du vecteur de probabilité P . Pour le calcul du nombre moyen de produits finis que l'on note \bar{X}_2^+ il faut multiplier chaque valeur possible de la variable aléatoire X_2^+ (de $X_2^{\max} = 5$ à 0) par sa probabilité. Pour le cas étudié dans cette section, la valeur moyenne \bar{X}_2^+ est donc donné par 5 fois l'unique composante de P_0 , plus 4 fois la somme des composantes de P_1 , plus 3 fois la somme des composantes de P_2 , plus 2 fois la somme des composantes de P_3 , et plus 1 fois la somme des composantes de P_4 . En utilisant R on peut exprimer ce nombre moyen de produits finis de la façon suivante :

$$\bar{X}_2^+ = 5P_0 + \sum_{i=1}^5 \left((5-i)[1,1]({}^tR)^{i-1} P_1 \right)$$

Celui de produits intermédiaires est donné par la formule suivante :

$$\bar{X}_1 = \sum_{i=1}^{\infty} \left([0,1]({}^tR)^{i-1} P_1 \right) = [0,1](Id - {}^tR)^{-1} P_1$$

avec Id la matrice identité. $[1,1]$ et $[0,1]$ sont des vecteurs ligne.

Par analogie avec le développement en séries de $1/(1-x)^2$, le nombre moyen de demandes en attente peut aussi être simplifié :

$$\bar{X}_2^- = \sum_{i=1}^{\infty} \left(i.[1,1]({}^tR)^{i+4} P_1 \right) = [1,1](Id - {}^tR)^{-2} ({}^tR)^5 P_1$$

Un calcul numérique donne alors pour $b=4$, $h=2$, $\lambda=1$, $\mu_1=3$ et $\mu_2=2$ le coût moyen suivant :

$$\begin{aligned} \bar{c} &= \bar{X}_1 + h\bar{X}_2^+ + b\bar{X}_2^- = 0.4999990692 + 2 \times 2.226867201 + 4 \times 1.226823400 \\ &= 9,861027071 \end{aligned}$$

7.1.4.2 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$

Le calcul du coût moyen, pour la politique kanban généralisé exposée sur la figure 7.3, s'obtient par une formule similaire. Pour $b=4$, $h=2$, $\lambda=1$, $\mu_1=3$ et $\mu_2=2$ le résultat obtenu est :

$$\begin{aligned} \bar{c} &= \bar{X}_1 + h\bar{X}_2^+ + b\bar{X}_2^- = 1,166666667 + 2 \times 3,711683405 + 4 \times 0,08744097927 \\ &= 8,939797394 \end{aligned}$$

La seconde méthode de calcul que nous allons exposer permet aussi d'effectuer des calculs exacts mais les valeurs numériques pourront être beaucoup plus précises puisqu'il est parfois possible d'obtenir des valeurs rationnelles exactes. De plus, il ne faut plus effectuer les itérations nécessaires à l'obtention de la matrice R pour atteindre ensuite de façon précise la valeur du coût moyen.

7.2 Autre méthode pouvant donner des solutions exactes en nombres rationnels

L'idée initiale de cette méthode a été de chercher la probabilité des états par déductions successives à partir d'un état de la chaîne de Markov. Prenons par exemple la chaîne de Markov de la politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$ (voir figure 7.2). Si la probabilité de l'état global défini par $X = (5, 0)$ est connue alors on peut en déduire celle de $X = (4, 1)$, puis on en déduit celle de $X = (4, 0)$ et ainsi de suite (voir figure 7.7).

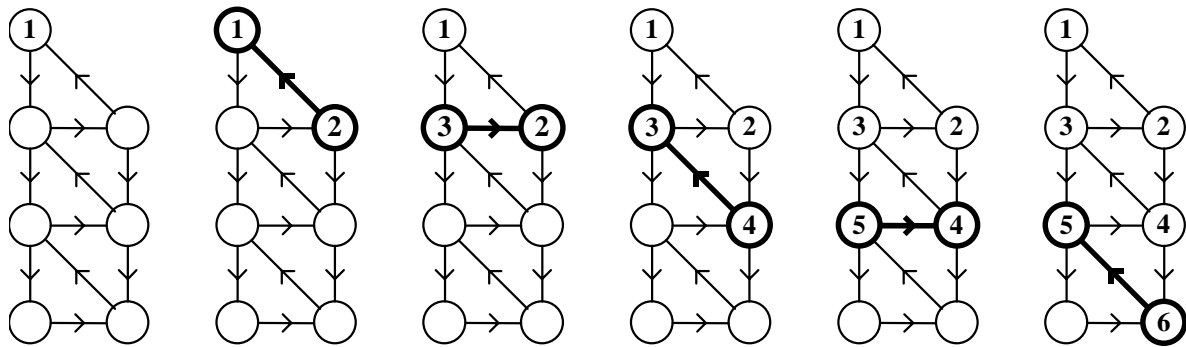


Figure 7.7 : Séquence de déduction de la probabilité des états d'un kanban généralisé ($K_2 = 1$, $S_1 = 0$ et $S_2 = 5$) sur le graphe d'événement de la chaîne de Markov.

L'ordre de déduction de la probabilité des états fournit une certaine numérotation des états. Cette numérotation va donner au générateur infinitésimal la structure requise pour cette seconde méthode de calcul de la chaîne de Markov.

7.2.1 Méthode de numérotation des états

La méthode de numérotation des états utilisée dans la section 7.1.1 nous donne une matrice de transition formée de trois diagonales de sous-matrices (A_i , B_i et C_i). La diagonale inférieure est constituée des sous matrices C_i qui ne sont pas inversibles. Or pour trouver des solutions fractionnaires avec la méthode que nous avons développée, il nous faut une diagonale inférieure constituée de sous-matrices inversibles. Ceci s'obtient en changeant la numérotation des états. Cette numérotation peut s'illustrer facilement dans un plan où les abscisses et les ordonnées sont respectivement représentées par X_2 et $(X_1 + X_2)$. Une telle représentation des états dans le plan n'est pas nouvelle [RYZ 1993]. Cette numérotation des

états s'effectue alors selon des segments où la valeur $X_1 + 2X_2$ est constante, on parcourt chaque segment en commençant par ceux dont la valeur constante est la plus grande. La figure 7.8 illustre cette numérotation pour un kanban généralisé qui est identique à celui de la figure 7.3, les segments sont représentés par des ovales. Ainsi le segment numéro 1 est constitué de l'état 1, le segment numéro 2 est constitué des états 2 et 3, le segment numéro 3 est constitué de l'état 4, le segment numéro 4 est constitué des états 5 et 6, et ainsi de suite.

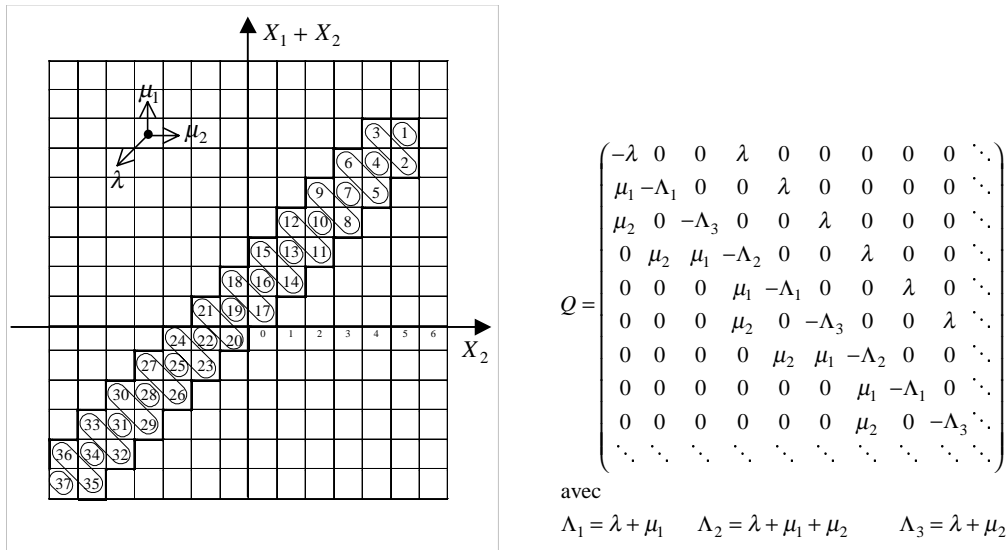


Figure 7.8 : Nouvelle numérotation des états pour un kanban généralisé ($K_2 = 1$, $S_1 = 1$ et $S_2 = 5$) avec son générateur.

Le générateur obtenu figure 7.8 a une structure répétitive. On veut définir des blocs de sous-matrices permettant d'obtenir deux diagonales de sous-matrices. De plus, à partir d'un certain rang, on souhaite que les sous-matrices de la diagonale inférieure soient inversibles. Pour obtenir de telles sous-matrices il faut effectuer deux partitions, une pour les lignes et une pour les colonnes du générateur. Les sous-matrices sont alors à l'intersection des sous-ensembles de lignes et de colonnes. La structure obtenue est alors une de celle illustrée par la formule (7.6) suivante :

$$Q = \begin{pmatrix} F_0 & 0 & 0 & 0 & 0 & 0 & \ddots \\ E_1 & F_1 & 0 & 0 & 0 & 0 & \ddots \\ 0 & E_2 & F_2 & 0 & 0 & 0 & \ddots \\ 0 & 0 & \ddots & \ddots & 0 & 0 & \ddots \\ 0 & 0 & 0 & E_i & F_i & 0 & \ddots \\ 0 & 0 & 0 & 0 & E_i & F_i & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \text{ ou } Q = \begin{pmatrix} E_0 & F_0 & 0 & 0 & 0 & 0 & \ddots \\ 0 & E_1 & F_1 & 0 & 0 & 0 & \ddots \\ 0 & 0 & E_2 & F_2 & 0 & 0 & \ddots \\ 0 & 0 & 0 & \ddots & \ddots & 0 & \ddots \\ 0 & 0 & 0 & 0 & E_i & F_i & \ddots \\ 0 & 0 & 0 & 0 & 0 & F_i & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \quad (7.6)$$

A partir d'un rang i les sous-matrices E_i et F_i ne changent plus car, à cause du nombre borné de produits dans le stock intermédiaire et l'encours de la station, le graphe d'événement de la chaîne de Markov devient répétitif. Pour mieux définir ces sous-matrices nous utilisons le terme « segment » déjà défini précédemment. On peut rappeler qu'un segment représente un ensemble d'états ergodiques où X_1+2X_2 reste constant. On numérote les segments en les classant suivant la valeur de X_1+2X_2 , le segment numéro un est celui où X_1+2X_2 a la plus grande valeur possible.

On définit ici k comme étant le plus petit entier tel que les segments successifs $k, k+1, k+2$ et $k+3$ comportent exactement $2X_1^{\max} + 2$ états. La sous-matrice E_i est alors située sur les lignes associées aux états des segments $k+1, k+2, k+3$ et $k+4$, et sur les colonnes associées aux états des segments $k, k+1, k+2$ et $k+3$. Conformément à la formule (7.6), la sous-matrice F_i se situe sur les lignes associées aux états des segments $k+1, k+2, k+3$ et $k+4$, et sur les colonnes associées aux états des segments $k+4, k+5, k+6$ et $k+7$.

Les sous-matrices E_j pour $j \leq i$ se situent sur des lignes correspondant aux états des quatre segments successifs $(k+4j-4i)+1, (k+4j-4i)+2, (k+4j-4i)+3$ et $(k+4j-4i)+4$, et sur des colonnes correspondant aux états des segments $(k+4j-4i), (k+4j-4i)+1, (k+4j-4i)+2$ et $(k+4j-4i)+3$. Conformément à la formule (7.6) et toujours avec la même valeur k , la sous-matrice F_j pour $j \leq i$ se situe sur des lignes correspondant aux états des quatre segments successifs $(k+4j-4i)+1, (k+4j-4i)+2, (k+4j-4i)+3$ et $(k+4j-4i)+4$, et sur des colonnes correspondant aux états des segments $(k+4j-4i)+4, (k+4j-4i)+5, (k+4j-4i)+6$ et $(k+4j-4i)+7$.

Pour trouver précisément la valeur de l'indice i il faut utiliser la structure indiquée sur la formule (7.6) et rester conforme à celle-ci. Ainsi, si $k+3$ est multiple de 4, on se retrouve dans le cas de la formule (7.6) situé à gauche. On a alors $i=(k+3)/4$ et $(k+0-4i)+4=1$ donc les lignes de la matrice F_0 correspondent à un unique segment. Si $k+3$ n'est pas multiple de 4 alors il faut considérer la formule (7.6) situé à droite. On a alors i qui est égal à la partie entière de $(k+3)/4$ et la sous-matrice E_0 a la particularité d'être constituée des lignes correspondant aux segments 1 à $(k-4i+4)$ et des lignes correspondant aux segments 1 à $(k-4i+3)$.

Prenons par exemple le cas de la figure 7.8 . On a alors $X_1^{\max} = 2, k=1$ et $i=1$. Les regroupements d'états nécessaires à la définition des lignes et des colonnes de chaque sous-matrice sont illustrés sur la figure 7.9. Les sous-matrices du kanban généralisé (avec $K_2=1, S_1=1$ et $S_2=5$) peuvent alors être obtenues à partir du générateur comme cela est montré figure 7.10.

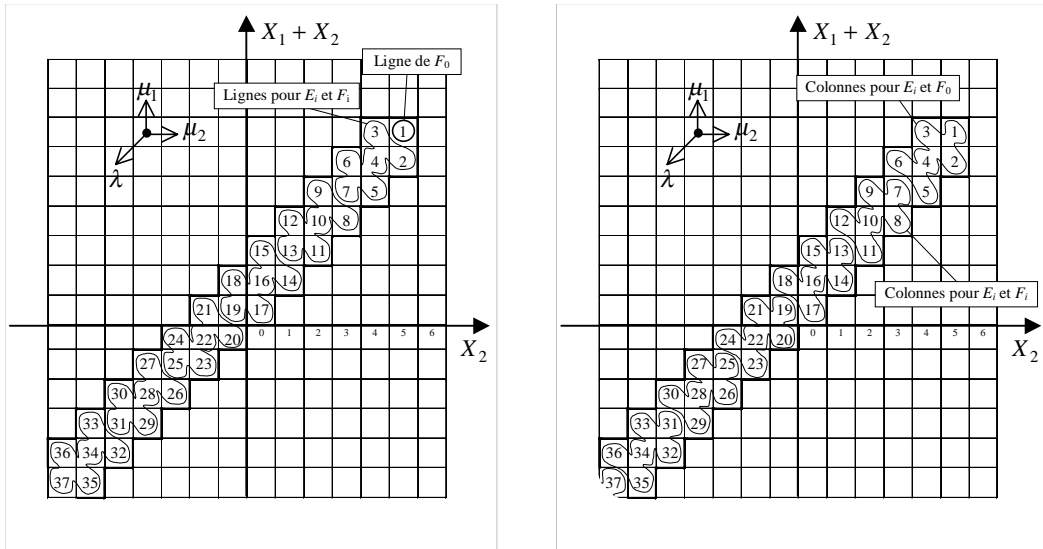


Figure 7.9 : Regroupement des états pour définir les lignes et les colonnes de chaque sous-matrice pour un kanban généralisé ($K_2 = 1$, $S_1 = 1$ et $S_2 = 5$).

$$Q = \begin{pmatrix} \begin{bmatrix} -\lambda & 0 & 0 & \lambda & 0 & 0 \\ \mu_1 & -\Lambda_1 & 0 & 0 & \lambda & 0 \\ \mu_2 & 0 & -\Lambda_3 & 0 & 0 & \lambda \\ 0 & \mu_2 & \mu_1 & -\Lambda_2 & 0 & 0 \\ 0 & 0 & 0 & \mu_1 & -\Lambda_1 & 0 \\ 0 & 0 & 0 & \mu_2 & 0 & -\Lambda_3 \\ 0 & 0 & 0 & 0 & \mu_2 & \mu_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ -\Lambda_2 & 0 & 0 & \lambda & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \dots \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} \mu_1 & -\Lambda_1 & 0 & 0 & \lambda & 0 \\ \mu_2 & 0 & -\Lambda_3 & 0 & 0 & \lambda \\ 0 & \mu_2 & \mu_1 & -\Lambda_2 & 0 & 0 \\ 0 & 0 & 0 & \mu_1 & -\Lambda_1 & 0 \\ 0 & 0 & 0 & \mu_2 & 0 & -\Lambda_3 \\ 0 & 0 & 0 & 0 & \mu_2 & \mu_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ -\Lambda_2 & 0 & 0 & \lambda & 0 & 0 \end{bmatrix} \dots \end{pmatrix}$$

avec

$$\Lambda_1 = \lambda + \mu_1 \quad \Lambda_2 = \lambda + \mu_1 + \mu_2 \quad \Lambda_3 = \lambda + \mu_2$$

Figure 7.10 : Sous-matrice sur le générateur d'un kanban généralisé ($K_2 = 1$, $S_1 = 1$ et $S_2 = 5$).

Pour le cas du kanban généralisé avec $K_2 = 3$, $S_1 = 2$ et $S_2 = 5$ on trouve $X_1^{\max} = 5$, $k = 3$ et $i = 1$. Les regroupements d'états nécessaires à la définition des lignes et des colonnes de chaque sous-matrice sont illustrés sur la figure 7.11. Les sous-matrices ne sont pas illustrées, E_i a une taille 12 par 12 dans ce cas.

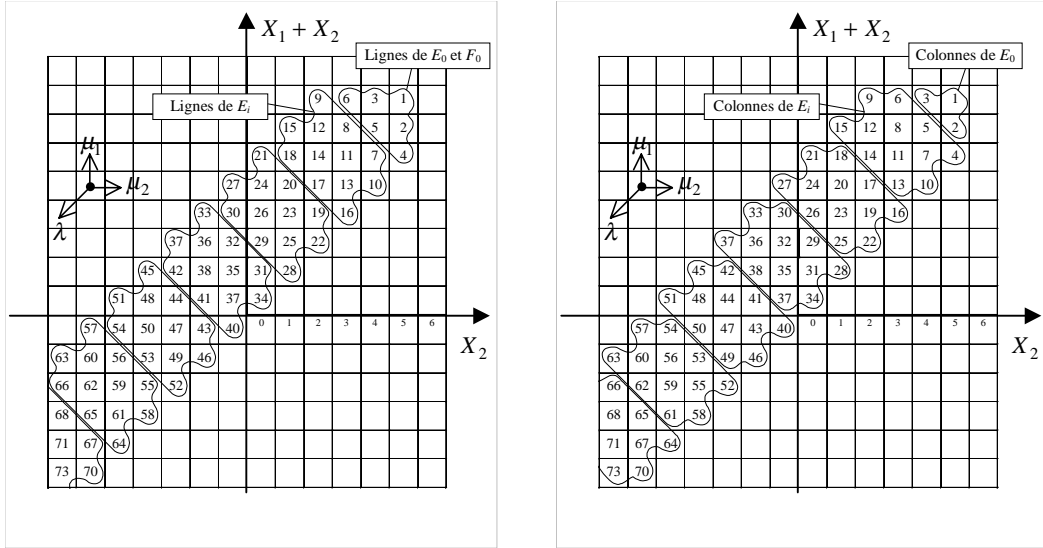


Figure 7.11 : Etats utilisés pour les lignes et les colonnes des sous-matrices (exemple pour le kanban généralisé avec $K_2 = 3$, $S_1 = 2$ et $S_2 = 5$).

On peut généraliser et automatiser l'obtention de ces sous-matrices lorsque la politique est exprimée par la formulation canonique (voir (4.30)) suivante :

$$F_1(t) = C_{(1,0)} \oplus C_{(1,1)} / X_1(t) \oplus C_{(1,2)} / X_1(t) / X_2(t)$$

$$F_2(t) = C_{(2,1)} \oplus C_{(2,2)} / X_2(t) \oplus X_1$$

Les paramètres $C_{(1,0)}$ et $C_{(2,1)}$ ne doivent pas être nuls pour permettre l'entrée de pièces dans les processus de fabrication, il n'ont pas d'autre influence sur la politique. Avec cette formulation canonique, nous avons utilisé plusieurs formules pour définir le générateur (7.6) et ses sous-matrices. Ces formules ont servi à automatiser l'obtention des générateurs infinitésimaux dans un code informatique, elles sont décrites ci-dessous.

Chaque segment est composé de plusieurs états, les états aux extrémités d'un segment l sont notés $X^B(l)$ lorsque $(X_1 + X_2)$ est minimal et $X^H(l)$ lorsque $(X_1 + X_2)$ est maximal. On peut calculer les coordonnées respectives $(X_1^B(l), X_2^B(l))$ et $(X_1^H(l), X_2^H(l))$ de $X^B(l)$ et $X^H(l)$. Elles sont données par les formules suivantes :

$$X_2^B(l) = \min \left(C_{(2,2)}, \frac{(C_{(2,2)} + C_{(1,2)} - l)}{2} + \frac{1 - (-1)^{(l - C_{(1,2)} + C_{(2,2)})}}{4} \right)$$

$$X_1^B(l) = C_{(1,2)} + C_{(2,2)} + 1 - 2X_2^B(l) - l$$

$$X_2^H(l) = \max(C_{(2,2)} - l + 1, X_2^B(l + C_{(1,1)} - 1))$$

$$X_1^H(l) = C_{(1,2)} + C_{(2,2)} + 1 - 2X_2^H(l) - l$$

On peut alors définir la numérotation du $c^{\text{ième}}$ élément d'un segment l par :

$$n(l, c) = c + \sum_{k=1}^{l-1} (X_2^B(k) - X_2^H(k) + 1)$$

Les numéros des états après des transitions λ , μ_1 et μ_2 partant d'un état numéro $n(l,c)$ sont respectivement donnés par les formules suivantes :

$$n_\lambda(l,c) = n(l+2,c) - X_2^B(l) + X_2^B(l+2) + 1$$

$$n_{\mu_1}(l,c) = n(l,c) + X_2^H(l-1) - X_2^B(l) - 1$$

$$n_{\mu_2}(l,c) = n(l,c) + X_2^H(l-1) - X_2^B(l) - 2$$

La position de la première sous-matrice E_i se situe entre les colonnes $n(\tilde{l},1)$ et $(n(\tilde{l}+4,1)-1)$ et entre les lignes $n(\tilde{l}+1,1)$ et $(n(\tilde{l}+5,1)-1)$ avec $\tilde{l} = \max(C_{(1,1)} - C_{(1,2)} + C_{(2,2)}, C_{(1,2)} - C_{(2,2)})$.

Pour le générateur d'un kanban généralisé où $K_2=1$, $S_1=1$ et $S_2=5$ (figure 7.8), on a alors $C_{(2,2)}=5$, $C_{(1,2)}=6$ et $C_{(1,1)}=2$. Ainsi $\tilde{l}=1$, le rang i vaut 1 et l'on obtient les sous-matrices suivantes :

$$F_0 = [-\lambda \ 0 \ 0 \ \lambda \ 0 \ 0], E_1 = \begin{bmatrix} \mu_1 - \Lambda_1 & 0 & 0 & \lambda & 0 \\ \mu_2 & 0 & -\Lambda_3 & 0 & 0 & \lambda \\ 0 & \mu_2 & \mu_1 & -\Lambda_2 & 0 & 0 \\ 0 & 0 & 0 & \mu_1 & -\Lambda_1 & 0 \\ 0 & 0 & 0 & \mu_2 & 0 & -\Lambda_3 \\ 0 & 0 & 0 & 0 & \mu_2 & \mu_1 \end{bmatrix} \text{ et } F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 & 0 \\ -\Lambda_2 & 0 & 0 & \lambda & 0 & 0 \end{bmatrix} \quad (7.7)$$

Pour le générateur de la figure 7.2, il suffit de prendre $C_{(2,2)}=5$, $C_{(1,2)}=5$ et $C_{(1,1)}=1$. Comme précédemment, $\tilde{l}=1$, le rang i vaut 1 et l'on obtient les sous-matrices F_0 , E_1 et F_1 qui suivent :

$$F_0 = [-\lambda \ 0 \ \lambda \ 0], E_1 = \begin{bmatrix} \mu_2 - \Lambda_3 & 0 & \lambda \\ 0 & \mu_1 & -\Lambda_1 & 0 \\ 0 & 0 & \mu_2 & -\Lambda_3 \\ 0 & 0 & 0 & \mu_1 \end{bmatrix} \text{ et } F_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ -\Lambda_1 & 0 & \lambda & 0 \end{bmatrix} \quad (7.8)$$

Les structures de générateur que nous utiliserons pour le calcul des probabilités stationnaires sont définies.

7.2.2 Méthode de calcul des probabilités

Comme pour la méthode précédente on pourra vérifier avec (7.4) si le système a un comportement stationnaire. Nous montrons qu'il est possible de trouver des solutions en nombre rationnel aux problèmes exposés sur les figures 7.2 et 7.3. Pour cela il faut utiliser un générateur qui est d'une forme identique à (7.6). La solution est alors très proche de celle donnée par la méthode de la matrice géométrique. On a :

$$P_j = P_i R^{j-i} \text{ pour } j > i$$

avec R solution de :

$$RE_i + F_i = 0$$

donc :

$$R = -F_i E_i^{-1}$$

Les sous-vecteurs P_j pour j allant de 0 à i sont donnés par le système d'équations suivant :

$$\begin{cases} {}^t P_j F_j + {}^t P_{j+1} E_{j+1} = 0 & \text{pour } j \text{ allant de } 0 \text{ à } i-1 \\ \sum_{j=0}^{i-1} {}^t P_j e + \sum_{j=i}^{\infty} ({}^t P_i R^{j-i}) e = 1 \end{cases} \quad (7.9)$$

La matrice R existe car E_i est inversible. Par contre R a une valeur propre égale à un. En effet pour la méthode de Neuts on trouve un vecteur π tel que ${}^t \pi A_i + {}^t \pi B_i + {}^t \pi C_i = 0$. A quelques permutations près le générateur est le même pour la seconde méthode, donc on peut trouver un autre vecteur π tel que ${}^t \pi E_i + {}^t \pi F_i = 0$ et ainsi on a ${}^t \pi (Id - (-F_i E_i^{-1})) = 0$.

Pour les problèmes exposés sur les figures 7.2 et 7.3, on a toujours obtenu une multiplicité égale à un pour la valeur propre 1 de R . On peut expliquer cela par le rang de R qui dans les deux cas vaut 3. Il ne reste alors au plus que deux valeurs propres différentes de un. Le vecteur P_i ne peut être qu'élément de la somme des sous-espaces caractéristiques relatifs à ces dernières valeurs propres. Si la multiplicité de la valeur propre un est trois alors il n'y a pas de solution. Si sa multiplicité est deux alors P_i est un vecteur propre de R associé à une valeur propre α , on a alors une solution de la forme ${}^t P_i = \alpha^i \cdot {}^t P_1$. Cette solution est celle d'un processus de naissance et de mort ce qui n'est pas le cas ici. On a ainsi toujours obtenu une multiplicité égale à un.

A cause de cette valeur propre un, il n'est pas possible de calculer $\sum_{j=0}^{\infty} R^j$. Cependant

$\sum_{j=i}^{\infty} ({}^t P_i R^{j-i})$ existe car ${}^t P_i$ fait partie d'un hyperplan h_1 constitué de la somme des sous-espaces caractéristiques relatifs aux valeurs propres positives et strictement inférieures à un. Dans ce plan h_1 l'application linéaire ${}^t R$ est stable (pour tout $v \in h_1$, ${}^t R v \in h_1$) et la restriction de ${}^t R$ à h_1 donne une matrice dont le rayon spectral est strictement inférieur à un pour les cas exposés figures 7.2 et 7.3. Il faut trouver cet hyperplan puis restreindre tout calcul à celui-ci.

Cet hyperplan est défini par l'ensemble des vecteurs orthogonaux aux vecteurs de base du sous-espace caractéristique relatif à la valeur propre un pour R . Si cette base est (u_1, \dots, u_k) alors h_1 est défini par :

$$\{x \mid {}^t u_1 x = 0, {}^t u_2 x = 0, \dots, {}^t u_k x = 0\}$$

L'espace vectoriel décrit par l'hyperplan h_1 correspond donc au noyau de la matrice dont les vecteurs ligne sont ${}^t u_1, {}^t u_2, \dots$ et ${}^t u_k$.

On peut alors définir une matrice N dont les lignes sont formées des vecteurs de base de ce noyau. En augmentant N des k vecteurs ligne issus de la base du sous-espace caractéristique relatif à la valeur propre 1 pour tR on définit la matrice de changement de base M . La matrice N' est définie comme étant identique à la matrice M^{-1} à laquelle on aurait retiré les k derniers vecteurs colonne. Les matrices N et N' effectuent simultanément un changement de base et une projection dans l'hyperplan h_1 . Le système (7.9) peut ainsi être remplacé par le suivant :

$$\begin{cases} {}^tP_j F_j + {}^tP_{j+1} E_{j+1} = 0 \text{ pour } j \text{ allant de } 0 \text{ à } i-1 \\ \sum_{j=0}^{i-1} {}^tP_j e + {}^tP_1 N' (Id - NRN')^{-1} Ne = 1 \end{cases} \quad (7.10)$$

Avec ${}^tP_1 N'$ on projette tP_1 dans la base de l'hyperplan h_1 puis une fois les calculs effectués dans cette base (NRN' est la restriction de R à cette base) on revient au repère initial avec N . Ce système de dimension finie peut être résolu pour donner les valeurs de P_j pour j allant de 0 à i . Pour j supérieur à i il suffit de calculer $P_j = P_i N' (NRN')^{j-i} N$.

Dans la section 7.2.3 qui suit, cette méthode est illustrée sur des exemples de petite taille. On abordera également section 7.2.5, des cas plus complexes dans lesquels une ou plusieurs valeurs propres supérieures à 1 peuvent apparaître dans la matrice R . Pour les cas simples considérés ci-dessous la matrice R n'a que des valeurs propres inférieures ou égales à 1, il est alors possible comme nous le montrons de calculer les probabilités de façon exacte.

7.2.3 Résultats numériques pour « des cas simples »

Pour un système constitué de deux mailles, nous considérons des cas tels que le nombre de pièces présentes dans le stock de produits intermédiaires et dans la station aval n'est pas supérieur à deux.

7.2.3.1 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$

Pour ce cas avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$ on a obtenu les résultats numériques suivants :

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1/2 & -1/2 & -1/1 & -5/6 \\ 0 & -1/3 & -2/3 & -2/3 \\ 2 & 2 & 7/2 & 17/6 \end{pmatrix}$$

On a $k = 1$ avec un vecteur propre :

$${}^t u_1 = (3/2 \quad 1 \quad 3/2 \quad 1)$$

qui permet de trouver les matrices :

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 10/11 & -1/11 & 3/11 \\ 0 & -1/11 & 10/11 & 3/11 \\ 0 & -1/11 & 10/11 & 3/11 \end{pmatrix}, M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 10/11 & -1/11 & 3/11 \\ 0 & -1/11 & 10/11 & 3/11 \\ 3/2 & 1 & 3/2 & 1 \end{pmatrix} \quad \text{et}$$

$$N' = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 11/3 & 11/3 \\ 3 & 8/3 & 14/3 \\ -9 & -23/3 & -32/3 \end{pmatrix}$$

Ces matrices permettent d'obtenir la probabilité de chaque état. Ces probabilités sont représentées figure 7.12.

7		
6		
5	1/6	
4	1/12	1/12
3	1/18	1/12
2	1/24	5/72
1	7/216	1/18
0	11/432	19/432
-1	13/648	5/144
-2	41/2592	71/2592
-3	97/7776	7/324
-4	17/1728	265/15552
-5	181/23328	209/15552
-6	571/93312	989/93312

Figure 7.12 : Valeur rationnelle de la probabilité de quelques états.

Les valeurs rationnelles obtenues peuvent être comparées aux valeurs numériques de la figure 7.4. Ceci a été réalisé sur la figure 7.13 qui montre les erreurs dues aux calculs numériques dans la première méthode. L'écart le plus important vaut 0,00000155103.

7		
6		
5	,00000155103	
4	,00000061317	,00000061317
3	,00000024641	,00000045081
2	,00000007028	,00000022055
1	,00000002909	,00000004442
0	,00000008679	,00000007198
-1	,00000011822	,00000014221
-2	,00000013235	,00000017976
-3	,00000013521	,00000019513
-4	,00000013094	,00000019598
-5	,00000012244	,00000018777
-6	,00000011168	,00000017427

Figure 7.13 : Ecart entre les valeurs rationnelles des probabilités obtenues figure 7.12 et les valeurs numériques des probabilités obtenues figure 7.4.

7.2.3.2 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$.

Pour ce cas avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$ on calcule R :

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1/7 & -2/7 & -2/7 & -5/21 & -1/2 & -17/42 \\ 1/7 & -3/14 & -3/14 & -1/7 & -3/7 & -5/14 \\ -2/21 & 1/7 & -4/21 & -4/21 & -2/7 & -1/3 \\ 6/7 & 12/7 & 12/7 & 9/7 & 19/7 & 15/7 \end{pmatrix}$$

On a $k = 1$ avec un vecteur propre ${}^t u_1 = (2/3 \ 3/2 \ 1 \ 2/3 \ 3/2 \ 1)$. Celui-ci permet de trouver les matrices :

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{27}{28} & \frac{-1}{28} & \frac{-1}{28} & \frac{5}{28} \\ 0 & 0 & \frac{-1}{28} & \frac{27}{28} & \frac{-1}{28} & \frac{5}{28} \\ 0 & 0 & \frac{-1}{28} & \frac{-1}{28} & \frac{27}{28} & \frac{5}{28} \end{pmatrix} \text{ et } N' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \frac{4}{11} & \frac{9}{11} & \frac{91}{55} & \frac{26}{55} & \frac{51}{55} \\ \frac{4}{11} & \frac{9}{11} & \frac{36}{55} & \frac{81}{55} & \frac{51}{55} \\ \frac{4}{11} & \frac{9}{11} & \frac{36}{55} & \frac{26}{55} & \frac{106}{55} \\ \frac{-20}{11} & \frac{-45}{11} & \frac{-169}{55} & \frac{-119}{55} & \frac{-244}{55} \end{pmatrix}$$

qui permettent d'obtenir les probabilités des états qui sont représentées figure 7.14.

7			
6			
5	11/210	11/30	
4	11/315	11/105	11/105
3	44/2205	11/252	11/140
2	11/980	11/490	143/2940
1	1177/185220	11/882	253/8820
0	671/185220	2629/370440	2057/123480
-1	1793/864360	1507/370440	33/3430
-2	4631/3889620	6061/2593080	14377/2593080
-3	12419/18151560	20911/15558480	473/148176
-4	8569/21781872	42097/54454680	22231/12101040
-5	103499/457419312	6919/15558480	57563/54454680
-6	266387/3557705760	224411/1524731040	266761/762365520

Figure 7.14 : Valeurs rationnelles de la probabilité de quelques états du problème de la figure 7.3 avec $\lambda = 1$, $\mu_1 = 3$ et $\mu_2 = 2$.

En comparant ces valeurs rationnelles exactes, à celles obtenues par la méthode de Neuts figure 7.5 on trouve neuf à dix chiffres significatifs égaux. En comparant les figures 7.4 et 7.12 on observe que la méthode de la matrice géométrique ne donne pas plus de cinq chiffres significatifs lorsqu'il y a une centaine d'itération dans le calcul de R . Cette différence de

vitesse de convergence est certainement due à une demande moins proche de la capacité de production en saturation dans le cas du kanban où $S_1=1$.

Outre le calcul des probabilités, nous calculons ci-dessous les coûts de ces systèmes respectifs.

7.2.4 Calcul du coût moyen et de quantités moyennes

Le coût instantané moyen est donné par $c(X) = X_1 + hX_2^+ + bX_2^-$.

7.2.4.1 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 0$ et $S_2 = 5$

En utilisant R , N et N' on peut exprimer le coût moyen et certaines quantités moyennes de l'exemple considéré. Ainsi, le nombre moyen de produits finis est donné par :

$$\bar{X}_2^+ = 5P_0 + \sum_{i=1}^2 ([6 - 2i, 6 - 2i, 5 - 2i, 5 - 2i] N ({}^t N' {}^t R {}^t N)^{i-1} {}^t N' P_1)$$

Celui de produits intermédiaires est donné par la formule suivante :

$$\bar{X}_1 = \sum_{i=1}^{\infty} ([1, 0, 1, 0] {}^t N ({}^t N' {}^t R {}^t N)^{i-1} {}^t N' P_1) = [1, 0, 1, 0] {}^t N (Id - {}^t N' {}^t R {}^t N)^{-1} {}^t N' P_1$$

Par analogie avec le développement en séries de $1/(1-x)^2$, le nombre moyen de demandes en attente peut aussi être simplifié :

$$\begin{aligned} \bar{X}_2^- &= \sum_{i=1}^{\infty} ([2i - 2, 2i - 2, 2i - 1, 2i - 1] {}^t N ({}^t N' {}^t R {}^t N)^{i-1} {}^t N' P_1) \\ &= [-2, -2, -1, -1] {}^t N (Id - {}^t N' {}^t R {}^t N)^{-1} {}^t N' {}^t R^2 P_1 + \\ &\quad [2, 2, 2, 2] {}^t N (Id - {}^t N' {}^t R {}^t N)^{-2} {}^t N' {}^t R^2 P_1 \end{aligned}$$

Un calcul donne alors pour $b=4$, $h=2$, $\lambda=1$, $\mu_1=3$ et $\mu_2=2$ le coût moyen suivant :

$$\begin{aligned} \bar{c} &= \bar{X}_1 + h\bar{X}_2^+ + b\bar{X}_2^- = 1/2 + h \times 481/216 + b \times 265/216 \\ &= 355/36 \\ &= 9.861111111 \end{aligned}$$

7.2.4.2 Politique kanban généralisé avec $K_2 = 1$, $S_1 = 1$ et $S_2 = 5$

Le calcul du coût moyen pour le politique kanban généralisé exposée sur la figure 7.3 s'obtient par une formule similaire. Pour $b=4$, $h=2$, $\lambda=1$, $\mu_1=3$ et $\mu_2=2$ le résultat obtenu est :

$$\begin{aligned} \bar{c} &= \bar{X}_1 + h\bar{X}_2^+ + b\bar{X}_2^- = 7/6 + h \times 343739/92610 + 4 \times 89077/1018710 \\ &= 3035687/339570 \\ &= 8.939797391 \end{aligned}$$

Les calculs exacts permettent de vérifier les premiers résultats numériques. La précision obtenue est presque identique à celle des calculs de probabilité, c'est-à-dire neuf chiffres exacts pour le coût du kanban généralisé et quatre pour celui du système de la figure 7.2.

Cette seconde méthode de calcul des probabilités présentées ici marchent bien pour des systèmes où X_1^{\max} n'est pas supérieur à deux. Au-delà les résultats obtenus ne sont pas aussi précis car des valeurs propres strictement supérieures à 1 apparaissent.

7.2.5 Résolution de cas plus complexes

L'exemple que nous étudions est celui d'une de politique kanban généralisé dont les paramètres sont $K_2 = 2$, $S_1 = 1$ et $S_2 = 5$. Pour une formulation canonique (4.30) le problème est défini par les paramètres $C_{(2,2)} = 5$, $C_{(1,2)} = 7$ et $C_{(1,1)} = 3$. Ce cas n'est plus considéré comme « simple » car $X_1^{\max} \geq 3$ (il suffit d'avoir $C_{(1,1)} \geq 3$). La résolution de cet exemple pourra facilement être étendue à toute autre politique exprimée par une courbe de commande, cependant il faut que la courbe de commande de la station 2 soit horizontale et que la courbe de commande de la station 1 devienne verticale à partir d'une certaine valeur de X_2 . Il n'est donc pas possible de traiter la politique basestock.

La méthode utilisée est celle qui a permis de trouver des solutions exactes rationnelles pour des problèmes considérés comme étant « simples ». Cependant, des modifications doivent être apportées à cette méthode car la matrice R possède en plus de la valeur propre un, d'autres valeurs propres qui sont strictement supérieures à un. Comme, il n'existe pas de méthode exacte pour trouver les valeurs propres d'une matrice dont le rang est de 8 ou plus, seules des approximations numériques restent possibles.

Pour l'exemple étudié il n'y a que deux valeurs propres supérieures ou égales à un. Ce sont les valeurs propres 1 et 46.91271779 qui sont de multiplicité un. Si le vecteur probabilité P_i ne se trouve pas dans la somme des sous-espaces caractéristiques relatifs aux valeurs propres positives et strictement inférieures à un, alors les vecteurs P_j vont croître exponentiellement en fonction de j .

La solution consiste donc, comme lorsqu'il n'y avait que la valeur propre un, à effectuer tous les calculs dans un hyperplan h_1 défini par la somme des sous-espaces caractéristiques relatifs aux valeurs propres positives et strictement inférieures à un. Cet hyperplan est orthogonal aux vecteurs propres u_1 et u_2 de R relatif aux valeurs propre 1 et 46,91271779 :

$$u_1 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & -5 & -2 \end{bmatrix}$$

$$u_2 = \begin{bmatrix} ,41e-11 & -,27e-11 & -,1451395901 & ,1630451114 & -,1129784225 & ,03396928169 & ,8496470703 & -,4653306468 \end{bmatrix}$$

On obtient une base pour cet hyperplan en calculant le noyau de la matrice carrée dont les deux premières lignes sont u_1 et u_2 tandis que les suivantes sont nulles. Les vecteurs de base disposés dans les lignes de cette matrice constituent alors la matrice N :

$$N = \begin{bmatrix} 0 & -0,5658959313 & 0,6867569096 & 0,3235625774 & 0,02827589265 & 0,1854787941 & 0,1632479946 & 0,2039171002 \\ 0 & 0,06535252367 & -0,2304612458 & -0,08283557348 & -0,0213923729 & 0,9458968755 & 0,04357503749 & 0,1966662479 \\ 0 & 0,7897021696 & 0,572937072 & 0,009113724499 & 0,1441837463 & 0,07227600232 & 0,1397385907 & 0,04990879591 \\ 0 & -0,09557242663 & -0,1469521609 & -0,02445045666 & 0,9732425412 & -0,02552927125 & 0,1361672732 & 0,04773714316 \\ -1 & 0,00 & -0,00000000000027 & 0,000000000001676 & -0,000000000000669 & 0,000000000000861 & 0,000000000021325 & -0,000000000042311 \\ 0 & 0,2067056778 & -0,303813902 & 0,8658801452 & -0,01667979221 & -0,07920180737 & -0,03775607086 & 0,327482499 \end{bmatrix}$$

En augmentant N des deux vecteurs propres de tR relatifs aux valeurs propres 1 et 46,91271779 on obtient une matrice de changement de base M :

$$M = \begin{pmatrix} 0, & -0,5658959313 & 0,6867569096 & 0,3235625774 & 0,02827589265 & 0,1854787941 & 0,1632479946 & 0,2039171002 \\ 0, & 0,06535252367 & -0,2304612458 & -0,08283557348 & -0,0213923729 & 0,9458968755 & 0,04357503749 & 0,1966662479 \\ 0, & 0,7897021696 & 0,572937072 & 0,009113724499 & 0,1441837463 & 0,07227600232 & 0,1397385907 & 0,04990879591 \\ 0, & -0,09557242663 & -0,1469521609 & -0,02445045666 & 0,9732425412 & -0,02552927125 & 0,1361672732 & 0,04773714316 \\ -1, & 0, & -0,000000000000027 & 0,0000000000001676 & -0,0000000000000669 & 0,0000000000000861 & 0,00000000000021325 & -0,00000000000042311 \\ 0, & 0,2067056778 & -0,303813902 & 0,8658801452 & -0,01667979221 & -0,07920180737 & -0,03775607086 & 0,327482499 \\ 1, & 9/4 & 3/2 & 27/8 & 1 & 9/4 & 3/2 & 27/8 \\ 0,02687631977 & -0,01818206369 & 0,05179065981 & -0,1309048845 & 0,1840835989 & -0,1245341531 & 0,354729038 & -0,8966049842 \end{pmatrix}$$

En ne prenant que les six premières colonnes de l'inverse de M on obtient une matrice qui est notée N' :

$$N' = \begin{pmatrix} 0,000000000008656 & 0,00000000000177 & -0,0000000000002267 & -0,0000000000011504 & -1, & 0,0000000000026906 \\ -0,5658959311 & 0,06535252378 & 0,7897021698 & -0,09557242659 & -0,000000000062365 & 0,206705678 \\ 1,020025642 & 0,09227611033 & 1,1281437 & 0,01807866815 & -0,1691336319 & 0,2196608445 \\ 0,7214224586 & 0,3719844967 & 0,5474057055 & 0,054732232 & -0,1566730191 & 1,590135351 \\ 0,3682851504 & 0,3151287119 & 0,6976252162 & 1,129314534 & -0,1678332823 & 0,5277477595 \\ 0,5562862094 & 1,345397243 & 0,6176522808 & 0,08960886847 & -0,1618918503 & 0,5609612802 \\ -1,477117688 & -1,516989262 & -2,64309747 & -0,7235142232 & 0,850679706 & -2,574377706 \\ -0,6209854831 & -0,7726499666 & -1,019032987 & -0,07184132685 & 0,3077162293 & -1,211742203 \end{pmatrix}$$

On trouve $\tilde{l} = 2$, avec E_i qui ne change plus à partir de $i = 1$. On cherche les vecteurs P_0 et P_1 en cherchant les solutions du système suivant :

$$\begin{pmatrix} {}^tP_0 & {}^tP_1 \end{pmatrix} \begin{pmatrix} E_0 & F_0 \\ 0 & E_1 \end{pmatrix} = 0$$

avec :

$$E_0 = \begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix}, F_0 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ et } E_1 = \begin{pmatrix} 3 & 0 & -4 & 0 & 0 & 0 & 1 & 0 \\ 2 & 3 & 0 & -6 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 3 & -6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 & -6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Le système a deux solutions :

$$\begin{bmatrix} {}^tP'_0 & {}^tP'_1 \end{bmatrix} = [[-63, -69/7, -117/7], [-2, -117/7, -4, -177/14, 0, -99/14, 1, -201/14]]$$

et

$$\begin{bmatrix} {}^tP''_0 & {}^tP''_1 \end{bmatrix} = [[243/2, 267/14, 225/7], [4, 225/7, 8, 663/28, 1, 363/28, 0, 639/28]]$$

Il faut trouver la bonne combinaison linéaire de ces deux vecteurs pour que l'on puisse obtenir un vecteur de probabilités. Ce vecteur doit être dans h_1 , ce qui signifie que sa projection dans cet hyperplan est égale à l'identité puisque le vecteur est déjà dans ce plan. Pour cela il suffit de trouver le noyau de la matrice suivante :

$${}^t(Id - N'N) \begin{pmatrix} P'_1 & P''_1 \end{pmatrix}$$

Une seule solution est trouvée, en la multipliant à $\begin{bmatrix} {}^tP'_0 & {}^tP'_1 \end{bmatrix}$ et $\begin{bmatrix} {}^tP''_0 & {}^tP''_1 \end{bmatrix}$ on trouve un vecteur probabilité $\begin{bmatrix} {}^tP'''_0 & {}^tP'''_1 \end{bmatrix}$ non normalisé :

$$\begin{bmatrix} {}^tP_0''' & {}^tP_1''' \end{bmatrix} = [-0,7941864589 \quad -0,6076741466] \begin{bmatrix} {}^tP_0' & {}^tP_1' \\ {}^tP_0'' & {}^tP_1'' \end{bmatrix}$$

Pour normaliser ce vecteur il suffit de le diviser par la somme des vecteurs probabilité avant normalisation. Cette somme est donnée par :

$$[1, 1, 1] P_0''' + [1, 1, 1, 1, 1, 1, 1, 1] {}^tN (Id - {}^tN {}^tR {}^tN)^{-1} {}^tN' P_1'''$$

On obtient ainsi les probabilités indiquées sur la figure 7.15.

7				
6				
5	0,0141536374	0,0631931264	0,3998909728	
4	0,01021080121	0,02830727481	0,1051557968	0,1051557968
3	0,00586818233	0,01334478371	0,03800905942	0,07306099181
2	0,003164571173	0,006630964055	0,01707844023	0,04143210417
1	0,001672428644	0,00339505117	0,00847364351	0,0222843449
0	0,0008783431642	0,001762572138	0,00436102342	0,01178913361
-1	0,0004605213237	0,0009204719616	0,002272676922	0,006202389835
-2	0,0002413882387	0,0004818710675	0,001189347906	0,003256811126
-3	0,0001265386333	0,000252515817	0,0006232948538	0,001708898599
-4	0,00006634268273	0,0001323831456	0,0003268039781	0,0008964368374
-5	0,00003478671224	0,00006941604924	0,0001713775039	0,0004701897865
-6	0,00001824181197	0,0000364020827	0,00008987650943	0,0002466064364

Figure 7.15 : Probabilités obtenues par la deuxième méthode avec $\lambda = 1$, $\mu_1 = 3$, $\mu_2 = 2$, $C_{(2,2)} = 5$, $C_{(1,2)} = 7$ et $C_{(1,1)} = 3$.

Pour calculer P_j pour $j > 1$ la formule :

$${}^tP_j = {}^tP_1 R^{j-1}$$

est exacte en théorie mais en pratique les erreurs d'arrondi croissent exponentiellement à cause de la valeur propre supérieure à un. Donc il vaut mieux utiliser la formule suivante :

$${}^tP_j = {}^tP_1 N' (NRN')^{j-1} N$$

qui donne les résultats de la figure 7.15 très proches de ceux de la figure 7.6.

7.3 Conclusion sur ces méthodes de résolution de la chaîne de Markov

Pour les systèmes de deux étages étudiés qui ont tous une valeur de X_1 bornée, la méthode de la matrice géométrique donne toujours de très bon résultats. La qualité des résultats est alors obtenue au prix d'un nombre suffisant d'itérations dans l'approximation d'une matrice R qui sert à calculer la plupart des composantes du vecteur de probabilité.

Nous avons aussi proposé une seconde méthode qui permet un calcul exact des probabilités en valeurs rationnelles, lorsque le stock de produits intermédiaires et la station 2 ont entre zéro et deux pièces. Les opérations sont plus complexes mais moins coûteuses en temps de

calcul car il n'y a pas d'itérations à effectuer. Lorsque X_1 peut être supérieur à 3 l'intérêt de cette seconde méthode est bien moindre car il est nécessaire de calculer toutes les valeurs propres. Ce calcul des valeurs propres n'étant pas exact, la méthode ne l'est pas non plus. Les résultats obtenus peuvent même être de moins bonne qualité qu'avec la méthode de la matrice géométrique.

Les deux méthodes que nous avons utilisées pourraient certainement être appliquées à des systèmes de production ayant trois mailles. Au delà de trois mailles, l'augmentation du nombre des états devrait rendre très vite ces méthodes inapplicables en pratique.

A partir de chacune de ces méthodes de calcul des probabilités il est aussi possible d'obtenir les coûts ou les valeurs moyennes de chaque file d'attente. Ceci peut être fait pour un très grand nombre de politiques, c'est-à-dire toute politique s'exprimant par une courbe de commande et telle que la courbe de commande de la station 2 soit horizontale et la courbe de commande de la station 1 soit verticale à partir d'une certaine valeur de X_2 , ce qui exclut le basestock.

Chapitre 8

Conclusion

8.1 Le bilan

Nous avons eu comme objectifs la comparaison et l'étude des politiques de pilotage des systèmes de production. Pour arriver à cette fin, nous avons calculé et comparé le coût de politiques optimales et sous-optimales en utilisant une technique de programmation dynamique. Cela nous a conduits à trouver et développer une formulation générale des politiques de gestion de production à flux tiré. Cette formulation a permis de trouver une technique capable de déceler toutes les identités de fonctionnement existant entre politiques. Elle a aussi été utilisée pour mettre au point des outils de calcul des coûts et de la probabilité de chaque état, dans le cas de systèmes composés de deux mailles et dont la politique de pilotage limite le nombre des produits intermédiaires. Nous avons ainsi essayé de développer des outils de calcul et de comparaison applicables quelle que soit la politique à flux tiré utilisée. Le développement d'autres techniques de calcul utilisant cette formulation devrait ainsi permettre de faciliter la comparaison et l'étude de la plupart des politiques existantes.

Nous avons commencé par faire quelques rappels sur les politiques de gestion de production dans le chapitre 2. Ensuite, dans le chapitre 3, nous avons introduit notre technique de formulation des politiques. Celle-ci consiste à déterminer, en fonction de variables d'état dites globales, le nombre de pièces présentes dans les processus de fabrication F_i . Pour montrer la validité de ce nouveau formalisme, nous avons montré qu'il permet d'exprimer totalement la dynamique d'un système de production. Il nous a alors été nécessaire de donner une définition de ce qui pourrait caractériser la dynamique des systèmes de production. Ainsi la validité de ce formalisme a pu être montrée pour des systèmes dont les processus de fabrication comportent une file d'attente suivie d'une station monoserveur et pour le cas de processus de fabrication plus généraux.

Nous avons en particulier insisté sur le rôle que pouvait avoir un stock tampon à l'intérieur d'un processus de fabrication. Lorsqu'il est absent, le processus de fabrication peut participer

au contrôle des flux de production en créant des blocages dus à une capacité d'accueil insuffisante. De plus, avec certains processus de fabrication, cette capacité d'accueil est imprévisible à partir des seules variables d'état globales, il n'est alors plus possible d'utiliser notre formulation directement.

En considérant qu'entre deux processus de fabrication, il n'y a pas de différence entre une pièce dans le stock de produits intermédiaires ou une pièce dans le stock tampon, on a remarqué qu'il peut être préférable d'utiliser une autre formulation des politiques. Cette formulation donne le nombre de pièces en cours de service dans les mécanismes de production M_i des processus de fabrication F_i sans compter celles qui sont présentes dans le stock tampon de F_i . Elle consiste donc à déterminer $M_i(X)$ au lieu de $F_i(X)$.

La validité de ces formulations des politiques étant démontrée, nous avons mis en œuvre ces formulations pour des lignes de dimension quelconque, gérées par des politiques basestock, kanban, kanban étendu ou kanban généralisé (chapitre 4). L'utilisation de l'algèbre (min,+) a facilité les calculs de $F_i(X)$. Pour toutes les politiques sous-optimales considérées nous avons pu exprimer $F_i(X)$ par une formule paramétrable. Cette formule a été appelée « formulation canonique ». L'algèbre (min,+) nous a également permis de démontrer qu'une classe très générale de politiques à flux tiré, modélisées par réseaux de files d'attente, pouvait être exprimée par cette formulation canonique.

Notre formulation des politiques a aussi été mise en œuvre pour des systèmes avec production par lots ou des systèmes ayant une station d'assemblage. Les formules obtenues diffèrent alors assez peu de la forme dite canonique. Nous nous sommes également intéressés au MRP, qui n'est pas une politique de gestion de production à flux tiré. Pour un certain type de MRP, une variante de notre formulation peut, dans certains cas, être utilisée. Cependant, dans le cas général, nous ne sommes pas parvenus à adapter notre formulation au MRP.

Chapitre 5 nous avons développé une technique permettant de trouver toutes les identités de fonctionnement pouvant exister entre deux politiques données. En particulier nous avons fait cette recherche entre les politiques kanban étendu et kanban généralisé. De nouvelles identités de fonctionnement ont été trouvées. Celles-ci varient suivant que l'on suppose les processus de fabrication quelconques ou composés d'une file d'attente et d'un monoserveur, suivant que les paramètres sont finis ou infinis. A l'exception de ces quelques identités de fonctionnement ces deux politiques sont différentes.

Dans le chapitre 6 nous utilisons une technique de programmation dynamique pour déterminer les politiques optimales de pilotage pour différents systèmes de production et calculer leur coût. Le coût de chaque politique optimale sert de référence à la comparaison des coûts entre politiques sous-optimales. Pour ces politiques sous-optimales, les méthodes de calcul du coût sont très proches des techniques de programmation dynamique utilisées dans le cas optimal. Au lieu d'optimiser une politique de pilotage, on fixe la politique de contrôle et

on trouve un coût pour un certain paramétrage. On optimise ensuite le paramétrage de la politique pour minimiser le coût.

Pour valider nos calculs nous avons commencé par reproduire les résultats obtenus par Veatch et Wein [VEA 1994]. Le système étudié est une ligne de deux étages où les processus de fabrication sont des stations exponentielles et où la demande est poissonnienne. En plus des politiques kanban, basestock et « fixed buffer » étudiées dans l'article, nous nous sommes particulièrement intéressés au kanban généralisé.

Nous avons ensuite considéré un système similaire pour lequel chaque demande peut, pendant un certain délai « exponentiel », ne pas être servie sans qu'il y ait pour autant des coûts supplémentaires. La satisfaction de la demande se fait au plus tôt car il est possible de servir la demande sans attendre la fin du délai associé. Il s'est avéré que plus le délai est important, plus le coût de la politique kanban généralisé est proche du coût de la politique optimale par rapport aux politiques kanban ou basestock. Les courbes de commande de la politique optimale semblent alors indiquer qu'il est optimal de ne pas faire de différence entre une demande avant ou après délai pour le pilotage de ce type de système, car lorsqu'une demande avant délai se transforme en demande après délai, le pilotage ne change pas.

Le cas d'un système où il y a obligation de livrer après la fin du délai associé à la demande est aussi étudié. Là encore l'intérêt du kanban généralisé augmente avec le délai. Cependant, lorsque ce délai augmente, le coût de la politique optimale est souvent bien moindre que celui des politiques sous-optimales étudiées.

Cela peut s'expliquer par l'aspect des courbes de commande. Celles-ci montrent que plus le délai augmente, plus le contrôle optimal considère différemment les demandes avant délai et celles après délai. Les contrôles sous-optimaux que nous avons choisis ne font par contre jamais la différence et ils prennent en compte la demande dès son arrivée sans attendre l'écoulement du délai associé.

Avec les mêmes procédés nous avons aussi étudié un système de deux étages produisant par lots, avec délai et livraison au plus tôt.

Enfin un système composé de deux mailles suivies d'une maille d'assemblage a été étudié. Les trois stations exponentielles n'ont pas permis d'introduire la présence d'un délai car celui-ci aurait ajouté une quatrième dimension à l'espace d'état, ce qui aurait trop augmenté les temps de calcul.

Ces méthodes de programmation dynamiques ne calculent pas le coût des politiques de façon très précise car elles utilisent un espace d'état tronqué. Nous avons donc dans le chapitre 7 développé deux techniques de calcul de la chaîne de Markov obtenue à partir de notre formulation des politiques. Elles sont inspirées de la méthode de la matrice géométrique de Neuts qui est utilisée pour des processus de quasi naissance et mort. Ces deux méthodes s'appliquent à un système de deux mailles avec monoserveur exponentiel et demande

poissonienne. Le nombre de produits intermédiaires doit être limité ; ces méthodes ne s'appliquent donc pas au basestock.

La première méthode donne de bons résultats qui dépendent du calcul d'une matrice par itérations successives et convergentes. La seconde méthode nécessite le calcul des valeurs propres supérieures ou égales à 1 dans une sous-matrice du générateur infinitésimal. Lorsque le nombre de produits intermédiaires n'est pas supérieur à deux, alors il y a une valeur propre 1 et aucune autre supérieure. Il est alors possible de trouver des solutions exactes en nombres rationnels. Lorsque le nombre de produits intermédiaires est supérieur à deux la méthode fonctionne mais les solutions sont moins précises. Ces deux méthodes permettent de calculer les probabilités de chaque état mais aussi un coût moyen proportionnel aux niveaux de stock et d'arriéré dans la livraison des demandes.

Des méthodes efficaces ont été obtenues pour comparer l'identité des politiques. Des comparaisons de coût entre politiques ont permis de se faire une idée sur leurs intérêts respectifs et cela pour différents systèmes de production. Des méthodes de calcul des probabilités et des coûts ont été obtenues. Ces résultats sont intéressants mais de nombreuses améliorations restent possibles.

8.2 Perspectives

La formulation que nous avons proposée semble être utilisable pour la plupart des systèmes et la plupart des politiques de gestion de production. Cependant son utilisation pour des politiques de gestion de production à flux poussé est moins adaptée car il faut modifier la formulation et respecter quelques hypothèses comme cela a été fait pour le MRP.

Avec les graphes d'événements discrets, il y a une notion de dualité entre l'utilisation de l'algèbre $(\min,+)$ et $(\max,+)$. Il serait intéressant de trouver dans l'algèbre $(\max,+)$ une formulation duale à celle que nous avons présentée. Cela devrait permettre d'obtenir une description du contrôle en fonction des temps de passage plutôt qu'en fonction de l'état du système à un instant donné, car une telle formulation duale devrait utiliser des dateurs plutôt que des compteurs.

Il serait intéressant de développer un algorithme automatisant la découverte d'identités entre politiques. Celui-ci ne devrait utiliser que de simples comparaisons entre polyèdres.

Nous avons remarqué, pour un système de deux étages avec un délai de livraison mais possibilité de satisfaction de la demande dès que possible, que la politique optimale ne fait en général pas de distinction entre une demande dans les délais ou hors délais. Cette propriété a été observée fréquemment par le calcul. Elle est peut être exacte et il serait intéressant de la démontrer de façon théorique.

On a trouvé une méthode exacte pour analyser un très grand nombre de politiques sur une ligne de deux étages. Il serait intéressant de trouver une méthode unique d'analyse des politiques qui serait applicable à la plupart des politiques sur des lignes de plusieurs étages, cette méthode serait alors forcément approximative. Une telle méthode commune simplifierait la comparaison et l'usage des différentes politiques à flux tiré.

Bibliographie

- [AXS 1993] – Axsater S., Rosling K. – « Installation vs. echelon stock policies for multilevel inventory control » – *Management Science*, 39, No.10, 1274-1280 (1993).
- [BAC 1992] – Baccelli F., Cohen G., Olsder G.J. and Quadrat J.P. – *Synchronization and Linearity : An Algebra for Discrete Event Systems* – John Wiley and Sons, New York, (1992).
- [BAY 2000] – Baynat B. – *Théorie des files d'attente : des chaînes de Markov aux réseaux à forme produit* – Paris : Hermès science, 326 p., (juin 2000).
- [BER 1991] – Berkley B., – «Tandem queues and kanban-controlled lines » – *International Journal of Production Research*, vol 29, no 10, p 2057-2081, (1991).
- [BER 1976] – Bertsekas D. P. – *Dynamic programming and stochastic control* – Academic Press, New York, (1976).
- [BER 1995] – Bertsekas D. P. – *Dynamic Programming and Optimal Control* – Vol. II. Athena Scientific, Belmont, MA, (1995).
- [BEL 1957] – Bellman R. E. – *Dynamic Programming*. – Princeton University Press, Princeton, New Jersey, (1957).
- [BEL 1958] – Bellman R. E. – « On a Routing Problem » – *Quarterly of Applied Mathematics*, vol. 16, p.87-90, (1958).
- [BOI 2000] – Boimond J.L., Lahaye S., Cottenceau B., Hardouin L. – « On Discrete Event Dynamic Systems in Dioids : Identification, Control, Extension of Timed Event Graphs » – *Conférence IAR'2000*, Nancy, France, (nov. 2000).
- [BOL 1999.a] – Bollon J-M., Di Mascolo M., Frein Y. – « Comparaison de politiques de gestion de production par rapport à l'optimal pour un système de deux stations livrant avec délai » – *Conférence JDA '99*, Nancy, France, (sept. 1999).
- [BOL 1999.b] – Bollon J-M., Di Mascolo M., Frein Y. – « Comparaison qualitative de deux politiques de gestion de production à flux tiré : Le kanban généralisé et le kanban étendu » – *Conférence MOSIM'99*, Annecy, France, (oct. 1999).
- [BOL 2000] – Bollon J-M., Di Mascolo M., Frein Y. – « Unified formulation of pull control policies using (min,plus) algebra » – *Conférence IAR'2000*, Nancy, France, (nov. 2000).

- [BON 1995] – Bonvik A. M., Couch C., Gershwin S.B. – « Improving a kanban controlled production line through rapid information dissemination » – *Conférence INFORMS*, New Orleans, (nov. 1995).
- [BON 1996] – Bonvik A. M. – « Performance Analysis of Manufacturing Systems Under Hybrid Control Policies » – Thèse Massachusetts Institute of Technology, (1996).
- [BUZ 1989.a] – Buzacott J. A. – « Queuing models of kanban and MRP controlled manufacturing Systems » – *Engineering Cost and Production Economics*, 17, 3-20, (1989).
- [BUZ 1989.b] – Buzacott J. A., Shanthikumar J.G. – « Stochastic Models of single stage MRP controlled production Systems » – note interne, University of Waterloo, (août 1989).
- [BUZ 1992] – Buzacott J. A., Price S., Shanthikumar J.G. – « Service Level in multistage MRP and basestock controlled production systems » – *Conference on New direction for Operation Research in Manufacturing*, (ed. G. Fandel, T. Gullledge and A.Jones), Springer, 445-463, (1992).
- [BUZ 1993] – Buzacott J. A., Shanthikumar J.G. – *Stochastic Models of Manufacturing Systems* – Editions Prentice-Hall, Englewood Cliffs, N.J., (1993).
- [BUZ 1994] – Buzacott J. A., Shanthikumar J.G. – « Safety stock versus safety time in MRP controlled production Systems » - *Management Science*, 40, No.12, p 1678-1689, (1994).
- [CHA 2000] – Chaouiya C., Liberopoulos G., Dallery Y. – « The extended kanban control system for production coordination of assembly systems » – , *IIE Transactions on Design and Manufacturing, Special Issue on Decentralized Control of Manufacturing Systems*, Vol. 32, No. 10, p. 999-1012, (2000).
- [CLA 1960] – Clark A.J., Scarf H. – « Optimal policies for a multi-echelon inventory problem » – *Management Science*, vol. 6, no. 4, p. 475-490, (1960).
- [COT 1999] – Cottenceau B. – « Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les diodes » – Thèse de Doctorat, Université d'Angers, (octobre 1999).
- [DAL 1995] – Dallery Y., Liberopoulos G. – « A new kanban type pull control mechanism for multi-stage manufacturing systems » – *Proceedings of 3rd European Control Conference*, Rome, Italie, (Sept. 1995).

- [DAL 2000] – Dallery Y., Liberopoulos G. – « Extended kanban control system: Combining kanban and base stock » , *IIE Transactions on Design and Manufacturing*, vol. 32, no. 4, p. 369-386, (2000).
- [DAV 1992] – David R., Alla H. – *Du Grafcet aux réseaux de Petri* – 2ème édition revue et augmentée, Editions Hermès 2-86601-325-5, (1992).
- [DIM 1996] – Di Mascolo M., Dallery Y. – « Performance evaluation of kanban controlled assembly systems » – *Symposium on Discrete Events and Manufacturing Systems of the Multiconference IMACS-IEEE/SMC CESA'96*, Lille, France (juil. 1996).
- [DIM 1990] – Di Mascolo M. – « Modélisation et évaluation de performances de systèmes de production gérés en kanban » – Thèse à l'Institut National Polytechnique de Grenoble, LAG, (fev. 1990).
- [DUR 1997] – Duri C. – « Etude comparative de gestions à flux tiré » – Thèse à l'Institut National Polytechnique de Grenoble, LAG, (janv. 1997).
- [ETT 1995] – Ettl M. – « Analysis and optimisation of kanban-controlled production lines » – Thèse – Friedrich Alexander University Erlangen Nuremberg Herausgeber, (1995).
- [FED.I 1986] – Federgruen A., Zipkin P. – « An Inventory Model With Limited Production Capacity And Uncertain Demands I. The Average-cost Criterion » – *Mathematics Of Operations Research* Vol.12, No.2, 193-207, (1986).
- [FED.II 1986] – Federgruen A., Zipkin P. – « An Inventory Model With Limited Production Capacity And Uncertain Demands II. The Discounted-cost Criterion » – *Mathematics Of Operations Research* Vol.12, No.2, 208-215, (1986).
- [FEN 1997] – Fenu R. – « Etude d'une politique de pilotage optimale d'un système de production » – Mémoire de DEA Automatique-Productique, LAG, (juin 1997).
- [FRE 1995] – Frein Y., Di Mascolo M., Dallery Y., « On the design of generalized kanban control systems » – *International Journal of Operations & Production Management*, Special Issue on "Modelling and Analysis of Just-In-Time Manufacturing Systems", Vol 15, n°9, pp. 158-184, (1995).
- [FU 1994] – Fu M., Hu J-Q. – « On the relationship of capacitated production/inventory models to manufacturing flow control models » – *Operations Research Letters* 18, No.1, 15-24, (1995).

- [GAU 1992] – Gaubert S. – « Introduction aux systèmes dynamiques à événements discrets » – Polycopié de cours ENSTA - ENSMP - Orsay (DEA ATS), 1992 (dernière mise à jour : 1999).
- [GLA 1994] – Glasserman P., Tayur S. – « The stability of a capacitated, multi-echelon production-inventory system under a base-stock policy » – *Operations Research*, 42, No.5, 913-925 (1994).
- [GLA 1995] – Glasserman P., Tayur S. – « Sensitivity analysis for base-stock levels in multiechelon production-inventory systems » – *Management Science*, 41, No.2, 263-281, (1995).
- [HAR 1995] – Hariharan R., Zipkin P. – « Customer-order information, leadtimes, and inventories » - *Management Science*, 41, No.10, 1599-1607, (1995).
- [KAR 1999] – Karaesmen F., Buzacott J., Dallery Y. – « Pull type control mechanism and advance information on customer orders » – *Proceedings of the Flexible Automation and Intelligent Manufacturing 99 Conference*, Tilburg, Netherlands, (juin 1999).
- [KAR 2000] – Karaesmen F., Dallery Y. – « A performance comparison of pull type control mechanisms for multi-stage manufacturing » – *International Journal of Production Economics*, vol. 68, no. 1, (oct. 2000).
- [KIM 1988] – Kimball G. – « General Principles of Inventory Control » – *Journal of Manufacturing and Operations Management*, vol 1, no 1, p 119-130, (1988).
- [LEE 1992] – Lee Y-J., Zipkin P. – « Tandem queues with a planned inventories » – *Operations Research*, 40, No.5, 936-947, (1992).
- [LIB 2000] – Liberopoulos, G., Dallery Y. – « A unified framework for pull control mechanisms in multi-stage manufacturing systems » – *Annals of Operations Research*, 93, 325-355, (2000).
- [LIP 1975] – Lippman, S. – « Applying a new device in the optimization of exponential queuing systems » - *Operations Research*, 23, 687-710, (1975).
- [MUC 1995] – Muckstadt J. A., Tayur S – « A comparison of alternative Kanban control mechanisms: Part 1. Background and Structural Results » – *IIE Transactions*, vol 27, no 2, p 140-150, (1995).
- [NEU 1981] – Neuts M. F. – *Matrix-geometric solutions in stochastic models : an algorithmic approach* – The John Hopkins University Press. Londres, R.U., (1981).

- [PUT 1994] – Puterman M. L. – *Markov decision processes: Discrete stochastic dynamic programming* – Wiley, New York, (1994).
- [ROS 1970] – Ross, S.M. – *Applied probability models with optimization applications* – Holden-Day Series in Management Sciences, San Francisco etc.: Holden-Day, Inc., (1970).
- [RYZ 1993] – Van Ryzin G., Lou S., Gershwin S. – « Production control for a tandem two-machine system » – *IIE Transactions*, vol. 25, no. 5, (1993).
- [SBI 1999] – Sbiti N., Di Mascolo M., Amghar M. – *Modélisation et propriétés du pilotage kanban généralisé pour les systèmes de production avec assemblage*. – Conférence MOSIM'99, Annecy France, (oct 1999).
- [SBI 2000] – Sbiti N. – « Evaluation de performances des systèmes de production à flux tiré contenant de l'assemblage » – Thèse de l'Université Mohamed V – Agdal présenté à l'Ecole Mohammadia d'Ingénieurs, (janv. 2000).
- [SCA 1960] – Scarf H. – « The Optimality of (s,S) Policies In The Dynamic Inventory Problem » – *Mathematical Methods in the Social Sciences*, 1959, Proc. 1st Stanford Symp. 196-202, (1960).
- [SUG 1977] - Sugimori Y., Kusunoki K., Cho F., and Uchikawa S. – « Toyota Production System and Kanban System, Materialisation of Just-in-time and Respect-for-human System » – *International Journal of Production Research*, 15(6), 553-564, (1977).
- [VEA 1992] - Veatch, M. H., Wein, L. M. – « Monotone control of queueing networks » – *Queueing Systems*, 12, No.3/4, 391-408, (1992).
- [VEA 1994] - Veatch M. H., Wein L. M. – « Optimal control of a two-station tandem production/inventory system » – *Operations Research*, 42, No.2, 337-350, (1994).
- [VER 2000] – De Véricourt F., Karaesmen F., Dallery Y. – « Dynamic scheduling in a make-to-stock system : a partial characterization of optimal policies » – *Operations Research*, volume 48, numéro 5, (Sept. – Oct. 2000).
- [WEB 1987] – Weber R.R., Stidham S., « Optimal control of service rates in networks of queues » – *Advances in Applied Probability*, 19, 202-218, (1987).
- [WEB 2001] – Weber R. – « Optimization and control » – Notes de cours – (<http://www.statslab.cam.ac.uk/~rrw1/oc/index.html>) (2001).

[ZIP 1995] - Zipkin P. – « Processing networks with planned inventories: Tandem queues with feedback » – *European Journal of Operations Research*, 80, No.2, 344-349, (1995).

[ZHE 1991] - Zheng Y-S., Federgruen A. – « Finding optimal (s,S) policies is about as simple as evaluating a single policy » – *Operations Research*, 39, No.4, 654-665, (1991).

[ZHA 1996] – Zhang W., Zipkin P. – « Production-inventory networks with constant processing times » – INFORMS Meeting, Washington, USA, (Mai 1996).

Annexe A : Notations

P_i	Stock de produits de l'étage i .
F_i	Processus de fabrication de l'étage i .
C_i	Capacité d'accueil d'un processus de fabrication F_i .
P'_i	Stock P_i plus stock tampon du processus de fabrication F_i .
M_i	Processus de fabrication de l'étage i privée de son stock tampon.
C'_i	Capacité d'accueil de M_i .
D_N	Nombre de demandes non servies.
$(X_1 \cdots X_N)$	Vecteur d'état dit global pour un système de production.
1_i	Vecteur d'état global qui est nul partout sauf pour la $i^{\text{ème}}$ composante égale à un.
$(e_{D_N}, e_{P_1} \cdots e_{P_N}, e_{F_1} \cdots e_{F_N}, e_{C_1} \cdots e_{C_N})$	Vecteur d'état pour un système de production (type (a)).
$(e_{D_N}, e_{P'_1} \cdots e_{P'_N}, e_{M_1} \cdots e_{M_N}, e_{C'_1} \cdots e_{C'_N})$	Vecteur d'état pour un système de production (type (b)).
E	Ensemble des vecteurs d'état admissibles avec un critère de type (a).
E'	Ensemble des vecteurs d'état admissibles avec un critère de type (b).
E_X	Ensemble des vecteurs d'état global admissibles associés à E
E'_X	Ensemble des vecteurs d'état global admissibles associés à E'
u_D	Événement d'arrivée d'une demande d'un produit fini.
u_{D^q}	Événement d'arrivée d'une demande de q produits finis.
u_{F_i}	Événement sortie d'une pièce de F_i avec C_i n'augmentant pas.
u_{FC_i}	Événement sortie d'une pièce de F_i avec C_i augmentant de un.
$u_{F^j C_i^k}$	Événement sortie de j pièces de F_i avec C_i augmentant de k .
u_{C_i}	Événement où C_i augmente de un.
u_{M_i}	Événement sortie d'une pièce de M_i avec C'_i n'augmentant pas.
$u_{MC'_i}$	Événement sortie d'une pièce de M_i avec C'_i augmentant de un.
$u_{M^j C'^k_i}$	Événement sortie de j pièces de M_i avec C'_i augmentant de k .
$u_{C'_i}$	Événement où C'_i augmente de un.
\emptyset	Événement nul.
U_e	Ensemble des événements pouvant se produire lorsque l'état est $e \in E$.
U'_e	Ensemble des événements pouvant se produire lorsque l'état est $e \in E'$.
EU	Ensemble des couples (e, u) avec $e \in E$ et $u \in U_e$.
$E'U'$	Ensemble des couples (e, u) avec $e \in E'$ et $u \in U'_e$.

- $\mathfrak{X}(e)$ Application qui pour un état $e \in E$ associe l'état global correspondant dans E_X .
- $\mathfrak{X}'(e)$ Application qui pour un état $e \in E'$ associe l'état global correspondant dans E'_X .
- $\mathfrak{E}(X)$ Application qui pour un état $X \in E_X$ associe l'état correspondant dans E .
- $\mathfrak{E}'(X)$ Application qui pour un état $X \in E'_X$ associe l'état correspondant dans E' .

Annexe B : Éléments annexes du chapitre 3

B.I Preuve d'un lemme et d'une proposition de la section 3.2.4.2 :

Preuve du lemme 3.1.b :

Pour un critère de type (a), on utilise des capacités d'accueil C_i infinies mais pour un critère de type (b) les capacités d'accueil C'_i observées peuvent varier et elles valent 0 ou 1.

On trouve $\mathfrak{X}'[E'] = E'_X$ par la définition 3.4.b. Montrons que $\mathfrak{X}'(\cdot)$ est la fonction inverse de $\mathfrak{E}'(\cdot)$ c'est-à-dire, pour tout élément e de E' , il nous faut montrer que $e = \mathfrak{E}'(\mathfrak{X}'(e))$:

- Avec (3.9) on trouve $e_D = -\min(0, e_{P'_N} - e_D)$ et $e_{P'_N} = -\min(e_D - e_{P'_N}, 0)$ donc

$$e_D = \mathfrak{E}'(\mathfrak{X}'(e))_D \text{ et } e_{P'_N} = \mathfrak{E}'(\mathfrak{X}'(e))_{P'_N}.$$

- Pour i allant de 1 à $N-1$ on trouve $\mathfrak{E}'(\mathfrak{X}'(e))_{P'_i} = e_{P'_i} + e_{M_{i+1}} - m_{i+1}(\mathfrak{X}'(e))$. D'après les hypothèses du lemme, $e_{M_{i+1}} = m_{i+1}(\mathfrak{X}'(e))$ donc $\mathfrak{E}'(\mathfrak{X}'(e))_{P'_i} = e_{P'_i}$.

- De même on trouve $\mathfrak{E}'(\mathfrak{X}'(e))_{M_i} = m_i(\mathfrak{X}'(e)) = e_{M_i}$ pour i allant de 1 à N .

- En utilisant (3.10), on obtient $\mathfrak{E}'(\mathfrak{X}'(e))_{C'_i} = 1 - m_i(\mathfrak{X}'(e)) = e_{C'_i}$ pour i allant de 1 à N .

On a donc $\mathfrak{E}'[E'_X] = E'$, ainsi E'_X et E' sont équipotents.

□

Preuve de la proposition 3.1.b :

Le lemme 3.1.b s'applique et il montre qu'il y a correspondance entre un état dit global de E'_X et un état de E' utilisé dans la définition 3.3.b. A chaque élément $X \in E'_X$ on va associer un élément $e = \mathfrak{E}'(X)$ et on obtient :

$$E' = \mathfrak{E}'[E'_X]$$

Les seuls événements possibles pour chacun des états sont déterminés par l'ensemble U'_e de la définition 3.1.b. On obtient alors :

$$EU' = \{(e, u) \mid e \in E'; u \in U'_e\}$$

Pour savoir s'il est possible de déterminer la dynamique il suffit de trouver une application D' définie de $E'U'$ vers E' qui vérifie les propriétés (3.9),(3.11),(3.13),(3.12) et de montrer qu'il n'en existe pas d'autres.

Suite à la définition de E' la propriété (3.9) est vérifiée car :

$$\min(e_D, e_{P'_N}) = \min(-\min(X_N, 0), -\min(-X_N, 0)) = 0$$

Puisque D' vérifie (3.11), (3.13) et (3.12) on en déduit ce qui suit :

- Pour tout couple $(e, \emptyset) \in EU'$ on a (3.11) c'est-à-dire $D'(e, \emptyset) = e$.

- Pour tout couple $(e, u_D) \in EU'$ on a $D'(e, u_D) = \mathfrak{E}'(\mathfrak{X}'(e) - 1_N)$.
- Pour les couples $(e, u_{MC'_i}) \in EU'$ avec $1 < i \leq N$ on trouve $D'(e, u_{MC'_i}) = \mathfrak{E}'(\mathfrak{X}'(e) + 1_i - 1_{i-1})$.
- Pour les couples $(e, u_{MC'_1}) \in EU'$ on obtient $D'(e, u_{MC'_1}) = \mathfrak{E}'(\mathfrak{X}'(e) + 1_1)$.

L'application D' est déterminée de façon unique. Tous les éléments nécessaires ont été trouvés de façon unique, donc on a bien déterminé la dynamique au sens de la définition 3.3.a. La proposition est démontrée. \square

B.II Dynamique pour un critère de type (a) généralisant les types de processus de fabrication utilisables.

Pour chaque état e de E , on note U_e tous les événements discrets pouvant se produire. A chaque instant n'y a qu'un événement possible, cependant dans le cas général certains événements peuvent correspondre à l'arrivée de plusieurs événements de base. On peut avoir une demande de q produits finis, une sortie de j pièces d'un processus de fabrication, une augmentation de k de la capacité d'accueil ou encore un mélange des deux derniers événements. La notation choisie pour chacun de ces événements respectifs est alors u_{D^q} , $u_{F^jC^0_i}$, $u_{F^0C^k_i}$ et $u_{F^jC^k_i}$ pour une maille i . Les événements u_D , u_{F_i} , u_{C_i} et u_{FC_i} utilisés précédemment correspondent respectivement à u_{D^1} , $u_{F^1C^0_i}$, $u_{F^0C^1_i}$ et $u_{F^1C^1_i}$. On a aussi $u_{F^0C^0_i} = u_{D^0} = \emptyset$. La figure B.1 illustre le fonctionnement d'une maille pour chacun de ces événements.

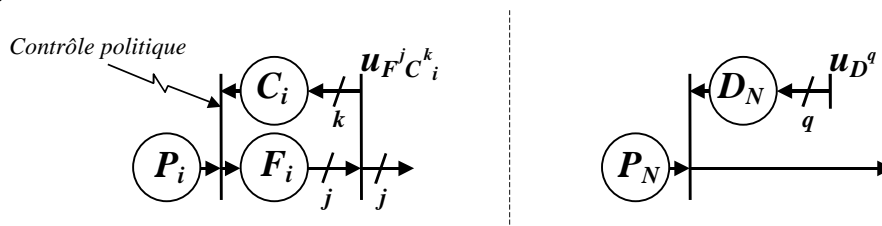


Figure B.1 : Réaction de la ligne aux événements $u_{F^jC^k_i}$ et u_{D^q} de U_e .

Voici la généralisation à différents types de processus de fabrication de la définition 3.3.a de la dynamique d'une ligne de production pour un critère de type (a).

Définition B.1.a :

La dynamique d'une ligne de production de n étages vue au niveau de ses stocks et processus de fabrication est définie par l'application D :

$$D : EU \longrightarrow E$$

$$(e, u) \mapsto f$$

avec

$$E \subset \mathbb{N}^{1+2N} \times (\mathbb{N} \cup \{\infty\})^N$$

$$U_e \subset \{\emptyset\} \cup \{u_{D^i} \mid i \in \mathbb{N}\} \cup \{u_{F^j C^k_i} \mid j \in \mathbb{N}, k \in \mathbb{N}, i \in \{1, \dots, N\}\}$$

$$EU = \{(e, u) \mid e \in E; u \in U_e\}$$

où e est l'état du système à l'instant qui précède l'arrivée d'un événement u et f est l'état du système après son arrivée. Pour tout e élément de E et pour tout $u_{F^j C^k_i}$ et u_{D^i} éléments de U_e , e et D doivent vérifier les propriétés suivantes :

$$\min(e_D, e_{P_N}) = 0 \quad (\text{B.1})$$

$$D(e, \emptyset) = e \quad (\text{B.2})$$

$$\mathfrak{X}(D(e, u_{D^q})) = \mathfrak{X}(e) - q \cdot 1_N \quad (\text{B.3})$$

$$\mathfrak{X}(D(e, u_{F^j C^k_1})) = \mathfrak{X}(e) + j \cdot 1_1 \quad (\text{B.4})$$

$$\forall i \in \{2, \dots, N\}, \quad \mathfrak{X}(D(e, u_{F^j C^k_i})) = \mathfrak{X}(e) + j(1_i - 1_{i-1}),$$

$$\forall i \in \{1, \dots, N\}, \quad D(e, u_{F^j C^k_i})_{C_i} + D(e, u_{F^j C^k_i})_{F_i} = e_{C_i} + e_{F_i} + k - j \quad (\text{B.5})$$

□

L'équation (B.1) montre qu'il n'est pas possible d'avoir à la fois des produits finis et des demandes en attente. Avec (B.2) on montre que l'état ne doit pas changer en présence d'un événement nul \emptyset . La formule (B.3) montre qu'une demande de q produits (l'événement u_{D^q}) ne fait varier que la composante X_N qui représente l'inventaire des produits finis. Cet inventaire décroît proportionnellement aux demandes (à cause du terme $-q \cdot 1_N$). En (B.4) on observe que l'événement $u_{F^j C^k_1}$ correspond à j pièces qui sortent de F_1 et qui rentrent dans P_1 (voir le terme $j(1_1 - 1_{i-1})$). On observe aussi que suite à l'événement $u_{F^j C^k_i}$, la valeur $F_i + C_i$ augmente de $k - j$ unités (voir terme $+k - j$ en (B.5)). Cela signifie que la capacité d'accueil augmente de k unités tandis que j pièces sortent de F_i . Des pièces peuvent passer de P_{i-1} à F_i au même instant mais cela ne va cependant pas modifier la variation de $F_i + C_i$ qui restera à $k - j$ unités.

Remarque :

Pour un système où les processus de fabrication sont composés d'un stock tampon et d'un monoserveur, une détermination de la dynamique par la définition 3.3.a est équivalente à la définition B.1.a à condition d'utiliser

$$U_e = \{\emptyset, u_D\} \cup \{u_{F_i} \mid e_{F_i} \geq 1, i \in \{1, \dots, N\}\}$$

$$\text{et } E \subset \mathbb{N}^{1+2N} \times (\{\infty\})^N.$$

Ainsi les capacités d'accueil sont infinies et les événements possibles seront u_{D^i} et $u_{F^1 C^0_i}$.

□

Remarque :

Pour un système où F_i est composé d'un stock tampon et d'une station servant des lots de taille Q_i , si la demande porte sur une pièce à la fois alors la dynamique définie par la définition B.1.a doit avoir les propriétés suivantes :

$$U_e = \{\emptyset, u_D\} \cup \left\{ u_{F_i^{Q_i, C_i^0}} \mid e_{F_i} \geq Q_i, i \in \{1, \dots, N\} \right\}$$

$$\text{et } E \subset \mathbb{N}^{1+2N} \times \{\infty\}^N.$$

Ainsi on ne produit en F_i que par lots de taille Q_i et il n'y a pas de service dans un processus de fabrication F_i si celui-ci ne contient pas au moins Q_i pièces. \square

B.III Dynamique pour un critère de type (b) généralisant les types de processus de fabrication utilisables.

Pour utiliser un critère de type (b) il doit exister un stock tampon au début de chaque processus de fabrication. Pour cette définition de la dynamique, des vecteurs d'état et des événements discrets différents sont utilisés.

Les vecteurs d'état seront composés des valeurs D_N , C'_i , M_i et P'_i pour i allant de 1 à N et l'ensemble des vecteurs d'état admissibles est noté E' . Pour chaque état e de E' on note U'_e tous les événements discrets pouvant se produire. On peut avoir une demande de q produits finis, une sortie de j pièces d'un processus de fabrication, une augmentation de k de la capacité d'accueil de M_i ou encore un mélange des deux derniers événements. La notation choisie pour chacun de ces événements respectifs est alors u_{D^q} , $u_{M^j C'^k}$, $u_{M^0 C'^k}$ et $u_{M^j C'^k}$ pour une maille i . Les événements u_D , u_{M_i} , $u_{C'_i}$ et $u_{M C'_i}$ déjà utilisés correspondent respectivement à u_{D^1} , $u_{M^1 C'^0}$, $u_{M^0 C'^1}$ et $u_{M^1 C'^1}$. On a aussi $u_{M^0 C'^0} = u_{D^0} = \emptyset$. La figure B.2 illustre le fonctionnement d'une maille et du stock de produits finis pour chacun de ces événements.

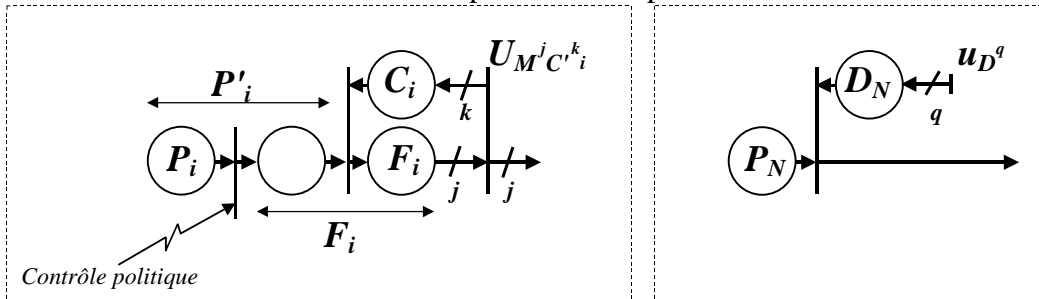


Figure B.2 : Réaction de la ligne aux événements $u_{M^j C'^k}$ et u_{D^q} de U'_e .

D'une façon formelle voici avec un critère de type (b) une définition de la dynamique d'une ligne production dont les processus de fabrication fonctionnent à flux poussé.

Définition B.1.b :

La dynamique d'une ligne de production de n étages, vue au niveau de ses stocks plus stocks tampon, et de ses processus de fabrication sans stock tampon, est définie par l'application D' :

$$D' : EU' \longrightarrow E'$$
$$(e, u) \mapsto f$$

avec

$$E' \subset \mathbb{N}^{1+2N} \times (\mathbb{N} \cup \{\infty\})^N$$

$$U'_e \subset \{\emptyset\} \cup \{u_{D^i} \mid i \in \mathbb{N}\} \cup \{u_{M^j C^k} \mid j \in \mathbb{N}, k \in \mathbb{N}, i \in \{1, \dots, N\}\}$$

$$EU' = \{(e, u) \mid e \in E; u \in U'_e\}$$

où e est l'état du système à l'instant qui précède l'arrivée d'un événement u et f est l'état du système après son arrivée. Pour tout e élément de E' et pour tout $u_{M^j C^k}$ et u_{D^i} éléments de U'_e , e et $D'(\cdot, \cdot)$ doivent vérifier les propriétés suivantes :

$$\min(e_D, e_{P_N}) = 0 \quad (\text{B.6})$$

$$D'(e, \emptyset) = e \quad (\text{B.7})$$

$$\mathfrak{X}'(D'(e, u_{D^i})) = \mathfrak{X}'(e) - q \cdot 1_N \quad (\text{B.8})$$

$$\mathfrak{X}'(D'(e, u_{M^j C^k})) = \mathfrak{X}'(e) + j \cdot 1_1$$

$$\forall i \in \{2, \dots, N\}, \quad \mathfrak{X}'(D'(e, u_{M^j C^k})) = \mathfrak{X}'(e) + j(1_i - 1_{i-1}), \quad (\text{B.9})$$

$$\forall i \in \{1, \dots, N\}, \quad D'(e, u_{M^j C^k})_{C'_i} + D'(e, u_{M^j C^k})_{M_i} = e_{C'_i} + e_{M_i} + k - j$$

□

On peut remarquer qu'une détermination de la dynamique par la définition 3.3.b est équivalente à la définition B.1.b, à condition d'utiliser $U'_e = \{\emptyset, u_D\} \cup \{u_{M^j C^k} \mid e_{M_i} \geq 1, i \in \{1, \dots, N\}\}$ et $E' \subset \mathbb{N}^{1+2N} \times (\{0, 1\})^N$.

Cette définition permet de comparer les dynamiques de différents systèmes. Les propriétés obtenues avec la définition B.1.b ou définition B.1.a seront relativement similaires.

B.IV Preuves et éléments de la section 3.3.3.1 :

Preuve du lemme 3.4.a :

On a $\mathfrak{X}[E] = E_X$ d'après la définition 3.4.a. Montrons que \mathfrak{X} est la fonction inverse de \mathfrak{e} .

Il faut donc montrer que pour tout élément e de E on a $e = \mathfrak{e}(\mathfrak{X}(e))$.

- Avec (B.1) on trouve $e_D = -\min(0, e_{P_N} - e_D)$ et $e_{P_N} = -\min(e_D - e_{P_N}, 0)$ donc

$$e_D = \mathfrak{e}(\mathfrak{X}(e))_D \text{ et } e_{P_N} = \mathfrak{e}(\mathfrak{X}(e))_{P_N}.$$

- Pour i allant de 1 à $N-1$ on trouve $\mathfrak{e}(\mathfrak{X}(e))_{P_i} = e_{P_i} + e_{F_{i+1}} - f_{i+1}(\mathfrak{X}(e))$. D'après les hypothèses du lemme $e_{F_{i+1}} = f_{i+1}(\mathfrak{X}(e))$ donc $\mathfrak{e}(\mathfrak{X}(e))_{P_i} = e_{P_i}$.
 - De même on a $\mathfrak{e}(\mathfrak{X}(e))_{F_i} = f_i(\mathfrak{X}(e)) = e_{F_i}$ pour i allant de 1 à N .
 - En utilisant $e_{C_i} = c_i(\mathfrak{X}(e))$ on trouve $\mathfrak{e}(\mathfrak{X}(e))_{C_i} = e_{C_i}$.
- Ainsi $\mathfrak{X}(\cdot) = \mathfrak{e}^{-1}(\cdot)$, E_X et E sont équipotents et $\mathfrak{e}[E_X] = E$. \square

lemme B.1.a :

Dans une ligne de production où il existe une dynamique selon la définition B.1.a, pour que les hypothèses du lemme 3.4.a soient vérifiées, il est nécessaire que l'ensemble U_e ait la propriété suivante :

- Quel que soit $e \in E$, si $u_{F^j C^k_i}$ est élément de U_e alors

$$k = c_i(\mathfrak{X}(e) + j(1_i - 1_{i-1})) + f_i(\mathfrak{X}(e) + j(1_i - 1_{i-1})) - e_{C_i} - e_{F_i} + j \quad (\text{B.10})$$
 et pour $i = 1$, $k = c_1(\mathfrak{X}(e) + j \cdot 1_1) + f_1(\mathfrak{X}(e) + j \cdot 1_1) - e_{C_1} - e_{F_1} + j$
- $u_{F^0 C^k_i}$ est élément de U_e si et seulement si $k = 0$.

\square

Preuve :

Quel que soit $e \in E$, si $u_{F^j C^k_i}$ est élément de U_e alors en utilisant (B.4) et le lemme 3.4.a on trouve $D(e, u_{F^j C^k_i}) = \mathfrak{e}(\mathfrak{X}(e) + j(1_i - 1_{i-1}))$ et $D(e, u_{F^j C^k_1}) = \mathfrak{e}(\mathfrak{X}(e) + j \cdot 1_1)$ pour $i = 1$. Ensuite, sachant que $\mathfrak{e}(X)_{C_i} = c_i(X)$ et $\mathfrak{e}(X)_{F_i} = f_i(X)$, en utilisant (B.5) on obtient le premier point du lemme. Le second point se déduit du premier en posant $j = 0$ et en remarquant que suite au lemme 3.4.a on a $\mathfrak{e}(\mathfrak{X}(e)) = e$.

\square

Preuve de la proposition 3.3.a :

Le lemme 3.4.a s'applique donc pour tout élément e de E on a $\mathfrak{e}(\mathfrak{X}(e)) = e$ et pour tout élément X de E_X on a $\mathfrak{X}(\mathfrak{e}(X)) = X$. On peut déterminer E à partir de E_X :

$$E = \mathfrak{e}[E_X]$$

L'ensemble U_e vérifie les propriétés du lemme B.1.a et $EU = \{(e, u) \mid e \in E; u \in U_e\}$.

Pour savoir s'il est possible de déterminer la dynamique il suffit de trouver une application D définie de EU vers E qui vérifie les propriétés (B.1),(B.2),(B.3),(B.4), (B.5) de la définition B.1.a et de montrer qu'il n'en existe pas d'autres.

Suite à la définition de E la propriété (B.1) est vérifiée car :

$$\min(e_D, e_{P_N}) = \min(-\min(X_N, 0), -\min(-X_N, 0)) = 0$$

Puisque D vérifie (B.2), (B.3), (B.4) et (B.5), on en déduit les seules valeurs qu'elle puisse prendre :

- Pour tout couple $(e, \emptyset) \in EU$ on a (B.2) c'est-à-dire $D(e, \emptyset) = e$.
- Pour tout couple $(e, u_{D^q}) \in EU$ en utilisant (B.3) et $\mathfrak{e}(\mathfrak{X}(e)) = e$ on trouve $D(e, u_{D^q}) = \mathfrak{e}(\mathfrak{X}(e) - q.1_N)$.
- Pour $(e, u_{F^j C^k_1}) \in EU$ en utilisant (B.4) on trouve $D(e, u_{F^j C^k_1}) = \mathfrak{e}(\mathfrak{X}(e) + j.1_1)$.
- Pour tout couple $(e, u_{F^j C^k_i}) \in EU$ avec $1 < i \leq N$ en utilisant (B.4) on trouve $D(e, u_{F^j C^k_i}) = \mathfrak{e}(\mathfrak{X}(e) + j.(1_i - 1_{i-1}))$.

Ainsi définie, l'application D est unique. Elle vérifie (B.2) et (B.3) car $\mathfrak{X}(D(e, u_{D^q})) = \mathfrak{X}(\mathfrak{e}(\mathfrak{X}(e) - q.1_N)) = \mathfrak{X}(e) - q.1_N$. En utilisant la même technique on montre $\mathfrak{X}(D(e, u_{F^j C^k_1})) = \mathfrak{X}(e) + j.1_1$ et $\mathfrak{X}(D(e, u_{F^j C^k_i})) = \mathfrak{X}(e) + j.(1_i - 1_{i-1})$.

Sachant que $\mathfrak{e}(\mathfrak{X}(e)) = e$ on trouve $D(e, u_{F^j C^k_i}) = \mathfrak{e}(\mathfrak{X}(e) + j.(1_i - 1_{i-1}))$ et $D(e, u_{F^j C^k_1}) = \mathfrak{e}(\mathfrak{X}(e) + j.1_1)$. En utilisant $\mathfrak{e}(X)_{C_i} = c_i(X)$, $\mathfrak{e}(X)_{F_i} = f_i(X)$ et le lemme

B.1.a, on montre que (B.5) est vérifié :

$$\begin{aligned} D(e, u_{F^j C^k_1})_{C_1} + D(e, u_{F^j C^k_1})_{F_1} &= c_1(\mathfrak{X}(e) + j.1_1) + f_1(\mathfrak{X}(e) + j.1_1) \\ &= e_{C_1} + e_{F_1} + k - j \end{aligned}$$

et

$$\begin{aligned} D(e, u_{F^j C^k_i})_{C_i} + D(e, u_{F^j C^k_i})_{F_i} &= c_i(\mathfrak{X}(e) + j.(1_i - 1_{i-1})) + f_i(\mathfrak{X}(e) + j.(1_i - 1_{i-1})) \\ &= e_{C_i} + e_{F_i} + k - j \end{aligned}$$

Tous les éléments nécessaires ont été trouvés de façon unique, et ils sont conformes à la définition B.1.a, ainsi la proposition est démontrée. □

Preuve du corollaire 3.2.a :

Un serveur infini ou un stock tampon infini permettent une capacité d'accueil infinie donc $c_i(X) = \infty$.

Pour n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i la capacité d'accueil dépend directement du nombre de pièces présentes en F_i , on trouve $c_i(X) = n_i.Q_i - f_i(X)$.

On a

$$U_e = \{\emptyset, u_D\} \cup \left\{ u_{F^{Q_i} C^{k_i}} \mid e_{F_i} \geq Q_i, i \in \{1, \dots, N\} \right\}$$

avec k_i nul si $c_i(X) = \infty$ et sinon en utilisant la formule (B.10) on trouve

$$k_i = c_i(\mathfrak{X}(e) + j.(1_i - 1_{i-1})) + f_i(\mathfrak{X}(e) + j.(1_i - 1_{i-1})) - e_{C_i} - e_{F_i} + j$$

et pour $i = 1$,

$$k_1 = c_1 (\mathfrak{X}(e) + j.1_1) + f_1 (\mathfrak{X}(e) + j.1_1) - e_{C_1} - e_{F_1} + j.$$

L'ensemble U_e est défini, les valeurs de F_i et C_i sont définies en fonction X donc en appliquant la proposition 3.3.a on démontre le corollaire.

□

B.V Section 3.3.3.2 : Détermination de la dynamique par M_i et C'_i en fonction de X pour un critère de type (b) .

Dans cette section, comme avec la proposition 3.3.a, nous allons définir une nouvelle façon de déterminer la dynamique. Cette fois-ci il s'agira d'une dynamique au sens de la définition B.1.b et elle sera déterminée à partir de deux applications définies de E'_X vers E dont la valeur est égale à M_i et C'_i .

Nous allons montrer, pour une ligne de production composée de processus de fabrication fonctionnant à flux poussé, qu'il est possible de déterminer la dynamique au sens de la définition B.1.b en sachant que M_i et C'_i sont égales à des applications définies de E'_X vers E' .

Un premier résultat consiste uniquement à montrer que l'existence de telles applications rendent E'_X et E' equipotents. Nous allons d'abord modifier et adapter la définition de l'application e' qui a précédemment été définie pour des systèmes comportant des processus de fabrication plus simples.

Définition 3.8.b :

On définit l'application e' par :

$$\begin{array}{ccc} e' : E'_X & \longrightarrow & E' \\ X & \mapsto & e \end{array}$$

avec

$$\begin{array}{lll} e_{D_N} = -\min(X_N, 0) & e_{M_i} = m_i(X) & \\ e_{P_N} = -\min(-X_N, 0) & e_{P'_i} = X_i - m_{i+1}(X) & e_{C'_i} = c'_i(X) \end{array}$$

où m_i et c'_i sont des applications définies respectivement de E'_X sur \mathbb{N} et de E'_X sur $\mathbb{N} \cup \{\infty\}$ pour i allant de 1 à N et vérifiant pour tout e de E' :

$$e_{M_i} = m_i(\mathfrak{X}(e)) \quad \text{et} \quad e_{C'_i} = c'_i(\mathfrak{X}(e)) \quad (2.11)$$

□

Pour rester cohérent avec la définition 3.5.a il suffit de remarquer que pour un système avec stock tampon et monoserveur on a, $c'_i(X) = 1 - m_i(X)$.

Lemme 3.4.b :

Pour une ligne de production, dont la dynamique est déterminée selon la définition B.1.b, si \mathfrak{e}' existe conformément à la définition 3.8.b alors E'_X et E' sont équipotents et $E' = \mathfrak{e}'[E'_X]$.

□

La preuve de ce lemme est une adaptation de celle utilisée pour le Lemme 3.4.a. En posant $c'_i(X) = 1 - m_i(X)$ on obtient le lemme 3.1.b. Comme pour le lemme 3.4.a, les hypothèses du lemme 3.4.b ne peuvent pas être vérifiées sans que l'ensemble U'_e ne vérifie certaines propriétés :

lemme B.1.b :

Dans une ligne de production où il existe une dynamique selon la définition B.1.b, pour que les hypothèses du lemme 3.4.b soient vérifiées, il est nécessaire que l'ensemble U'_e ait la propriété suivante :

- Quel que soit $e \in E'$, si $u_{M^j C^k_i}$ est élément de U'_e alors
$$k = c'_i(\mathfrak{X}'(e) + j(1_i - 1_{i-1})) + m_i(\mathfrak{X}'(e) + j(1_i - 1_{i-1})) - e_{C'_i} - e_{M_i} + j \quad (2.12)$$
et pour $i = 1$, $k = c'_1(\mathfrak{X}'(e) + j.1_1) + m_1(\mathfrak{X}'(e) + j.1_1) - e_{C'_1} - e_{M_1} + j$
- $u_{M^0 C^k_i}$ est élément de U'_e si et seulement si $k = 0$. □

La preuve de ce lemme est une adaptation de celle utilisée pour le Lemme B.1.a. Montrons à présent qu'il est possible de déterminer la dynamique au sens de la définition B.1.b en connaissant U'_e , E'_X et les applications m_i et c'_i .

Proposition 3.3.b :

Dans une ligne de production où il existe une dynamique selon la définition B.1.b, si \mathfrak{e}' existe conformément à la définition 3.8.b, alors on peut caractériser cette dynamique à partir de \mathfrak{e}' avec m_i , c'_i , E'_X et U'_e .

□

Cette proposition se prouve en adaptant la preuve de la proposition 3.3.a.

Corollaire 3.2.b :

Si pour une ligne de production la valeur M_i de chaque étage i est fonction de X , si les demandes arrivent une à une et si un des points suivants est vérifié pour chaque maille i alors on peut définir la dynamique de cette ligne de production selon la définition B.1.b.

- Le processus de fabrication est composé d'un stock tampon infini, suivi de n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i .
- Un stock tampon infini est présent en début de processus de fabrication suivi d'un serveur infini produisant par lots de taille Q_i .

Preuve :

Pour un serveur infini $c'_i(X) = \infty$ et pour n_i serveurs la capacité d'accueil dépend directement du nombre de pièces présentes en F_i , on trouve $c'_i(X) = n_i \cdot Q_i - f_i(X)$.

On pose $U_e = \{\emptyset, u_D\} \cup \{u_{M^i C^k_i} \mid e_{M_i} \geq Q_i, i \in \{1, \dots, N\}, k = k_i\}$ avec k_i nul si $c'_i(X) = \infty$ et sinon on utilise la formule (2.12) :

$$k_i = c'_i(\mathfrak{X}'(e) + j(1_i - 1_{i-1})) + m_i(\mathfrak{X}'(e) + j(1_i - 1_{i-1})) - e_{C'_i} - e_{M_i}$$

$$\text{et pour } i = 1, k_1 = c'_1(\mathfrak{X}'(e) + j.1_1) + m_1(\mathfrak{X}'(e) + j.1_1) - e_{C'_1} - e_{M_1}.$$

L'ensemble U_e est défini, les valeurs de M_i et C'_i sont définies en fonction X donc en appliquant la proposition 3.3.b on démontre le corollaire. \square

Remarquons que dans le corollaire 3.2.b il n'est pas possible de considérer un quelconque processus de fabrication avec stock tampon infini présent sur son entrée car c'_i n'est pas défini en fonction de X , par contre cela est possible dans le corollaire 3.2.a.

Pour des pièces servies une à une il suffit de fixer la taille des lots à un. Le corollaire 3.2.b montre que dans plusieurs cas avec un critère de type (b), la connaissance du nombre de pièces présentes dans les M_i en fonction de l'état global suffit à déterminer la dynamique du système. Pour les cas qui ne sont pas traité par le corollaire 3.2.b on peut utiliser la proposition 3.3.b qui permet de déterminer la dynamique d'une ligne de production relativement à un critère de type (b), à partir d'une application ε' et de l'ensemble des événements U'_e possibles pour chaque état e .

B.VI Section 3.3.3.3 : Relations entre les deux types de dynamique pour des processus de fabrication avec stock tampon et multiserveur.

On s'intéresse ici au cas particulier d'une ligne composée de processus de fabrication F_i ayant un stock tampon suivi d'un multiserveur. Ce dernier est composé de de n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i .

Dans toute cette section, les événements admissibles sont décrits par les définitions qui suivent. Pour un critère de type (a) on a :

Définition B.2.a :

Pour une ligne de production dont les processus de fabrication sont composés d'un stock tampon qui est suivi de n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i , on définit les événements discrets pouvant se produire dans un état e de E , par l'ensemble :

$$U_e = \{\emptyset, u_D\} \cup \left\{ u_{F_i, C_i^0} \mid e_{F_i} \geq Q_i, i \in \{1, \dots, N\} \right\}.$$

□

De même, pour un critère de type (b) on a :

Définition B.2.b :

Pour une ligne de production dont les processus de fabrication sont composés d'un stock tampon qui est suivi de n_i serveurs mis en parallèle et traitant les pièces par lots de taille Q_i , on définit les événements discrets pouvant se produire dans un état e' de E' , par l'ensemble :

$$U'_e = \{\emptyset, u_D\} \cup \left\{ u_{M_i, C_i^0} \mid e_{M_i} \geq Q_i, i \in \{1, \dots, N\} \right\}.$$

□

On montre dans cette section 3.3.3.3 qu'il est possible de trouver une dynamique pour la définition B.1.b si une dynamique pour la définition B.1.a existe. Pour cela il est nécessaire de connaître dans chaque maille i la valeur de F_i en fonction du vecteur d'état global. Cette section 3.3.3.3 ressemble beaucoup à la section 3.2.4.3 où se trouve la proposition 3.2. Le lemme 3.2 pourra même être utilisé sans aucune modification.

Définition B.3 :

L'application V est définie par

$$V : E \longrightarrow V[E]$$

$$e \mapsto e' = (e'_D, e'_{P_1}, \dots, e'_{P_N}, e'_{M_1}, \dots, e'_{M_N}, e'_{C_1}, \dots, e'_{C_N})$$

$$\text{où } e'_D = e_D, e'_{P_N} = e_{P_N}, e'_{M_i} = \min(n_i \cdot Q_i, e_{F_i}),$$

$$e'_{P_i} = e_{P_i} + e_{F_{i+1}} - e'_{M_i}, e'_{C_i} = n_i \cdot Q_i - e'_{M_i}$$

□

En prenant $n_i \cdot Q_i = 1$, on obtient la précédente définition de V (définition 3.6).

En général E et E' ne sont pas équipotents, cependant il est possible de trouver une application surjective définie de E vers E' .

Lemme B.2 :

Si E est connu et si les propriétés (3.1), (3.2), (3.3), $M_i + C_i' = n_i \cdot Q_i$ et $M_i = \min(n_i \cdot Q_i, F_i)$ sont vérifiées alors $E' = V[E]$.

□

Preuve :

Les propriétés (3.2), (3.3) sont vérifiées donc pour tout élément e de E il existe un élément e' de E' vérifiant $\mathfrak{X}(e) = \mathfrak{X}'(e')$ et réciproquement pour tout élément e' de E' il existe un élément e de E vérifiant $\mathfrak{X}(e) = \mathfrak{X}'(e')$. Avec la propriété (3.1) on montre que $e'_{D_N} = e_{D_N}$ et $e'_{P_N} = e_{P_N}$. Les propriétés $M_i = \min(n_i \cdot Q_i, F_i)$ et $M_i + C_i' = n_i \cdot Q_i$

permettent de montrer pour $i \in \{1, \dots, N\}$ que $e'_{M_i} = \min(n_i \cdot Q_i, e_{F_i})$ et $e'_{C_i} = n_i \cdot Q_i - e'_{M_i}$. On montre aussi que pour $i \in \{1, \dots, N-1\}$, $e'_{P_i} = e_{P_i} + e_{F_{i+1}} - e'_{M_{i+1}}$. Ainsi pour chaque élément e de E on a $V(e) \in E'$ et pour chaque élément e' de E' il existe un élément e de E tel que $e' = V(e)$. On a bien $E' = V[E]$.

□

Définition B.4 :

L'application W est définie par

$$W : EU \longrightarrow W[EU]$$

$$(e, u) \mapsto \left(V(e), u_{M_i, C_i^{Q_i}} \right) \text{ si } u = u_{F_i, C_i^0} \text{ et } (V(e), u) \text{ sinon}$$

□

Proposition B.1 :

Dans une ligne de production où l'application D définie de E vers EU détermine la dynamique au sens de la définition B.1.a, alors si la valeur F_i de chaque maille i est égale à une fonction f_i définie de E_X sur \mathbb{N} et si les relations (3.1), (3.2), (3.3), $M_i + C_i' = n_i \cdot Q_i$ et $M_i = \min(n_i \cdot Q_i, F_i)$ sont vérifiées, alors on en déduit qu'une unique dynamique au sens de la définition B.1.b, est déterminée par :

$$D' = V \circ D \circ W^{-1} \text{ définie de } E' = V[E] \text{ vers } EU' = W[EU]$$

□

Preuve :

Montrons que W est bijective. Par définition V et W sont surjectives. Supposons qu'il existe $W(e, u) = W(f, v)$ alors si $u = u_{F_i, C_i^0}$, $\left(V(e), u_{M_i, C_i^{Q_i}} \right) = \left(V(f), u_{M_i, C_i^{Q_i}} \right)$ et $v = u_{F_i, C_i^0} = u$, donc $u = v$ et $V(e) = V(f)$. On peut montrer facilement en utilisant la définition de V que e et f respectent $e_D = f_D$, $e_{P_N} = f_{P_N}$, $e_{P_i} + e_{F_{i+1}} = f_{P_i} + f_{F_{i+1}}$ et $\min(n_i \cdot Q_i, e_{F_i}) = \min(n_i \cdot Q_i, f_{F_i})$. On remarque alors que $\mathfrak{X}(e) = \mathfrak{X}(f)$ ainsi pour $i \in \{1, \dots, N\}$ on montre que $e_{F_i} = f_i(\mathfrak{X}(e)) = f_i(\mathfrak{X}(f)) = f_{F_i}$ et donc $e_{P_i} = f_{P_i}$. Les applications V et W sont donc bijectives ce qui montre que D' est bien définie.

On peut vérifier facilement que toutes les propriétés de la définition B.1.b sont vérifiées par l'application D' . On vient de montrer que la proposition exhibait une dynamique au sens de la définition B.1.b, montrons qu'il n'en existe pas d'autre.

En utilisant le lemme B.2 on trouve $E' = V[E]$. Pour toute application D' vérifiant les propriétés de la définition B.1.b et pour tout e de E tel que $e' = V(e)$ on obtient ce qui suit:

- Les ensembles U_e et U'_e sont respectivement définis en début de section par

$$U_e = \{\emptyset, u_D\} \cup \left\{ u_{F_i C_i^0} \mid e_{F_i} \geq Q_i, i \in \{1, \dots, N\} \right\} \text{ et}$$

$$U'_e = \{\emptyset, u_D\} \cup \left\{ u_{M_i C_i^{Q_i}} \mid e_{M_i} \geq Q_i, i \in \{1, \dots, N\} \right\}.$$

Avec $M_i = \min(n_i \cdot Q_i, F_i)$ on obtient $V(e)_{M_i} = e'_{M_i} \geq Q_i \Leftrightarrow e_{F_i} \geq Q_i$, on trouve alors

$$EU' = \{(e', u') \mid e' \in E', u' \in U'_e\} = \{W(e, u) \mid e \in E, u \in U_e\} = W[EU].$$

- Avec (B.7) on a $D'(e', \emptyset) = e' = V \circ D \circ W^{-1}(e, \emptyset)$.

- Avec (B.8) et (B.9) on obtient :

$$\mathfrak{X}'(D'(e', u_D)) = \mathfrak{X}'(e') - 1_N$$

$$\mathfrak{X}'\left(D'\left(e', u_{M^Q_1 C^Q_1}\right)\right) = \mathfrak{X}'(e') + Q_1 \cdot 1_1$$

$$\forall i \in \{2, \dots, N\}, \mathfrak{X}'\left(D'\left(e', u_{M^Q_i C^Q_i}\right)\right) = \mathfrak{X}'(e') - Q_i \cdot 1_{i-1} + Q_i \cdot 1_i$$

de même avec (B.3) et (B.4) on a :

$$\mathfrak{X}(D(e, u_D)) = \mathfrak{X}(e) - 1_N$$

$$\mathfrak{X}\left(D\left(e, u_{F^Q_1 C^Q_1}\right)\right) = \mathfrak{X}(e) + Q_1 \cdot 1_1.$$

$$\forall i \in \{2, \dots, N\}, \mathfrak{X}\left(D\left(e, u_{F^Q_i C^Q_i}\right)\right) = \mathfrak{X}(e) - Q_i \cdot 1_{i-1} + Q_i \cdot 1_i$$

On montre avec (3.2) et (3.3) que $\mathfrak{X}'(e') = \mathfrak{X}'(V(e)) = \mathfrak{X}(e)$ et on obtient alors ce qui suit :

$$\mathfrak{X}'(D'(e', u_D)) = \mathfrak{X}(D(e, u_D)) = \mathfrak{X}(D \circ W^{-1}(e', u_D)) \text{ et}$$

$$\forall i \in \{1, \dots, N\}, \mathfrak{X}'\left(D'\left(e', u_{M^Q_i C^Q_i}\right)\right) = \mathfrak{X}\left(D\left(e, u_{F^Q_i C^Q_i}\right)\right) = \mathfrak{X}\left(D \circ W^{-1}\left(e', u_{M^Q_i C^Q_i}\right)\right).$$

Supposons que nous ayons $\mathfrak{X}'(f') = \mathfrak{X}(f)$ alors avec (3.1), (3.2), (3.3), $M_i + C'_i = n_i \cdot Q_i$ et $M_i = \min(n_i \cdot Q_i, F_i)$ on montre que $f' = V(f)$. Ceci permet de modifier le précédent résultat, on trouve :

$$D'(e', u_D) = V \circ D \circ W^{-1}(e', u_D) \text{ et}$$

$$\forall i \in \{1, \dots, N\}, D'\left(e', u_{M^Q_i C^Q_i}\right) = V \circ D \circ W^{-1}\left(e', u_{M^Q_i C^Q_i}\right).$$

On a bien D' défini par $V \circ D \circ W^{-1}$, la proposition est démontrée. \square

Cette proposition nécessite la connaissance de l'application D qui détermine la dynamique au sens de la définition B.1.a et des fonctions f_i définies de E_X sur \mathbb{N} . Le corollaire qui suit atteint les mêmes objectifs sans qu'il soit nécessaire de connaître D . Il est suffisant de savoir que la dynamique existe selon la définition B.1.a.

Corollaire B.1 :

Pour une ligne de production, dont la dynamique existe selon la définition B.1.a, si pour tout état $e \in E$ les valeurs $F_i = e_{F_i}$ sont égales à $f_i(\mathcal{X}(e))$ où f_i est une fonction définie de E_X sur \mathbb{N} . Si les relations (3.1), (3.2), (3.3), $M_i + C'_i = n_i \cdot Q_i$ et $M_i = \min(n_i \cdot Q_i, F_i)$ sont vérifiées, alors on peut déterminer avec f_i et E_X la dynamique de cette ligne de production au sens de la définition B.1.b.

□

Preuve :

On utilise la proposition B.1 pour montrer qu'il existe une dynamique au sens de la définition B.1.b. Le lemme 3.2 permet de montrer que $E'_X = E_X$. En utilisant la proposition 3.3.b, $M_i + C'_i = n_i \cdot Q_i$ et $M_i = \min(n_i \cdot Q_i, F_i)$ on montre qu'il est possible de déterminer la dynamique de cette ligne de production au sens de la définition B.1.b avec seulement f_i et E'_X .

□

Résumé : Nous nous sommes intéressés dans cette thèse aux politiques de gestion de production à flux tiré. Après avoir constaté la diversité des politiques existantes (basestock, kanban, politiques hybrides dont kanban étendu et kanban généralisé, pilotage optimal, ...), nous avons proposé une formulation unifiée pour décrire leur dynamique. La validité de cette formulation est démontrée, puis nous l'appliquons à la description de plusieurs politiques connues. Un algorithme est donné pour pouvoir appliquer facilement cette formulation à une catégorie très générale de politiques. Cette formulation commune des politiques nous a permis de trouver les identités de fonctionnement qui existaient entre le kanban étendu et le kanban généralisé.

En introduisant des coûts de possession et des coûts de non satisfaction des demandes, nous avons aussi effectué une comparaison des coûts de plusieurs politiques sous-optimales, en utilisant à chaque fois la politique optimale et son coût comme référence. La politique optimale est calculée par programmation dynamique.

Pour finir nous proposons deux méthodes de calcul exact des probabilités stationnaires dans une ligne de production à deux étages où des politiques à flux tirées très générales sont utilisées. Les distributions obtenues sont utilisées pour calculer les coûts de possession et de non satisfaction des demandes.

Mots clés : flux tiré, basestock, kanban, kanban étendu, kanban généralisé, politique optimale, programmation dynamique, dioïde, algèbre min plus, chaîne de Markov.

Title: Study of different pull control policies for productions systems

Abstract: In this thesis we are interested in pull control policies. After having noted the diversity of the existing policies (basestock, kanban, hybrid policies like extended kanban and generalized kanban, optimal control policies, ...), we proposed a unified formulation to describe their dynamics. The validity of this formulation is shown, then we apply it to the description of several known policies. We give an algorithm which allows to easily apply this formulation to a very general category of policies. This common formulation of the policies permitted us to find some identities of behavior existing between the extended kanban and the generalized kanban.

We introduce an holding cost and a backorder cost, that allow us to make comparisons between the costs of several sub-optimal policies. Each time we used the optimal policy and its cost as a reference. The optimal policy is calculated by dynamic programming.

To finish, we propose two exact methods for the calculation of the stationary probabilities in a two stages production facility using a pull control policy. These probabilities are then used to make an exact calculation of the holding and backorder costs.

Key words: pull control, basestock, kanban, extended kanban, generalized kanban, optimal policy, dynamic programming, dioïd, min plus algebra, Markov process.