



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Projet de Semestre

**Modèles mathématiques
appliqués à la
gestion de production :
Les systèmes Kanbans**

Fournier Baptiste

Responsables:
Professeur Th. M. Liebling et Hêche Jean-François
ROSO-DMA

Hiver 1999 - 2000

Table des matières

1	Introduction	5
2	Généralités sur les méthodes Kanbans	6
2.1	Fonctionnement d'un système Kanban	6
2.2	Principaux critères décisionnels	8
3	Modèles mathématiques existants	10
3.1	Le modèle de Monden (1981) (Formule de Toyota)	10
3.2	Le modèle de Kimura et Terada (1981)	11
3.3	Le modèle de Bitran et Chang (1987)	12
3.4	Le modèle de Philipoom, Rees, Taylor et Huang (1990)	14
3.5	Le modèle de Wang et Wang (1990)	16
3.6	Le modèle de Bard et Golany (1991)	19
3.7	Conclusion	22
4	Développement d'un nouveau modèle	24
4.1	Politiques de production	25
4.2	Hypothèses et notations	26
4.3	Etats et matrice génératrice	28
4.4	Définition des transitions	29
5	Résultats et analyse du modèle	32
5.1	Comparaison avec le modèle de Wang et Wang	32
5.2	Influence du couple (NE,NU) dans le graphique	34
5.3	Comparaison des politiques	35
5.4	Evolution de (NE, NU) optimaux en fonction de K	38
6	Possibilités d'extension du modèle	39
7	Conclusion	42

TABLE DES MATIÈRES

A Codes utilisés dans <i>Matlab 5.3.0</i>	43
Bibliographie	65

Chapitre 1

Introduction

Au cours de ces dernières décennies, contraintes de résoudre les problèmes inhérents à la réalité économique, les entreprises ont ressenti le besoin de posséder des modèles de décision en gestion de production toujours plus performants. Cependant, même si des expressions telles *just in time production* ou *non-stock production* sont des plus répandues aujourd'hui, le développement de modèles mathématiques tenant compte des réels problèmes des entreprises est un domaine où, tout ou presque reste à faire. Ce projet a donc les ambitions suivantes :

- expliquer le fonctionnement d'un système Kanban et fournir une liste des principaux critères décisionnels pour l'évaluation d'un tel système ;
- établir un catalogue des modèles existants, proposés dans la littérature, en mettant en évidence les hypothèses nécessaires à leur développement, ainsi que les approches de modélisation utilisées ;
- développer un nouveau modèle de décision pour un système Kanban.

Chapitre 2

Généralités sur les méthodes Kanbans

2.1 Fonctionnement d'un système Kanban

L'approche *just in time (JIT)* comme contrôle de production à l'aide de kanbans a été la cible de nombreuses attentions au cours de ces dernières années. Cette idée est le fruit de M. Ohno qui fut le premier à l'appliquer pour Toyota, en s'inspirant des supermarchés américains, où l'on peut trouver ce que l'on veut, quand on veut et dans la quantité requise. M. Ohno l'a appliqué dans le but de :

- réduire le *work-in-progress (WIP)* et le temps de production ,
- augmenter la vitesse d'échange d'information entre deux postes de travail successifs ,
- augmenter la productivité.

En fait, le kanban est le nerf d'une production *JIT*; le kanban, traduction de *carte* en japonais, est en général une enveloppe attachée à un conteneur qui circule entre des postes de travail. Pendant longtemps, l'industrie s'est contentée, pour inventaire, de feuilles avançant avec les composantes tout au long de la production; ce type de système est généralement appelé *push-type system*. Inversément, dans un système *JIT*, qui est donc appelé *pull-type system*, le kanban agit comme un ordre de production, dont la fonction consiste à transporter des composantes en aval et d'émettre un ordre de production en amont. Les composantes sont supposées arriver au poste de

2.1. FONCTIONNEMENT D'UN SYSTÈME KANBAN

travail suivant au temps désiré et dans les quantités voulues. On distingue principalement deux systèmes de production Kanban:

- le système à un kanban
- le système à deux kanbans.

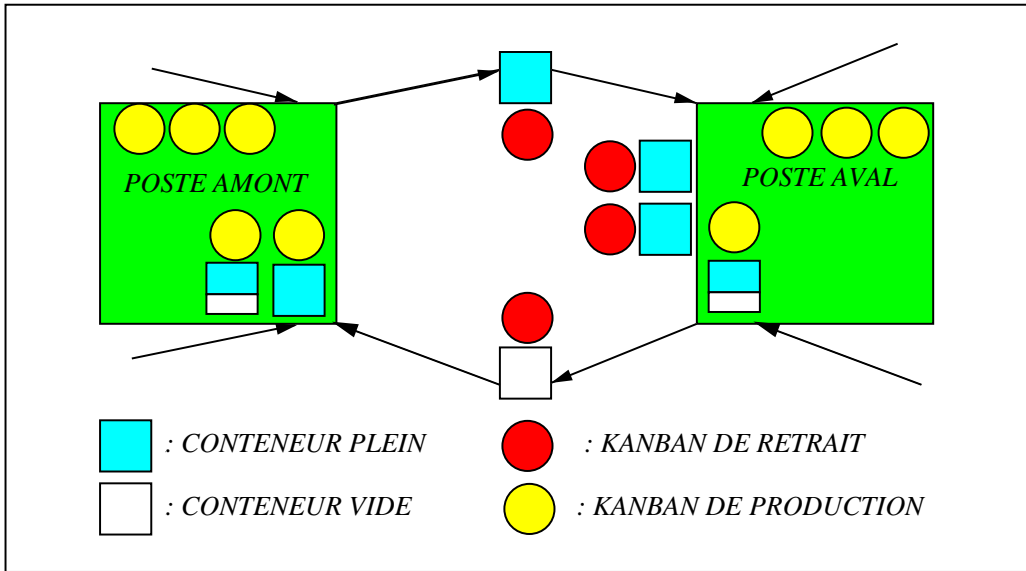


FIG. 2.1: *Système à deux kanbans*

Pour un système à un kanban, dès que des composants sont requises au i^e poste de travail, un kanban au $(i - 1)^e$ poste qui reçoit ainsi l'ordre de lancer sa production afin de remplir ce conteneur et de le renvoyer le plus rapidement possible au i^e poste.

Pour un système à deux kanbans (cf. figure 2.1), on distingue les deux types de cartes suivants:

- le kanban de retrait : qui circule, entre deux postes de travail successifs ,
- le kanban de production : qui reste au même poste de travail et agit comme ordre de production.

La manière, avec laquelle ces kanbans fonctionnent, est la suivante. Pour un i^e poste de travail, quand des composants sont requises au $(i + 1)^e$ poste (le i^e poste est informé par le kanban de retrait du $(i + 1)^e$ poste), le kanban de

production placé sur un conteneur plein est retiré et remplacé par le kanban de retrait du $(i + 1)^e$ poste. Le conteneur est ensuite envoyé au $(i + 1)^e$ poste où il sera traité. Le kanban de production retiré est mis de côté, en attendant la nouvelle production. Entre-temps, au i^e poste, la production va reprendre si il y a au moins un kanban de production et un conteneur, muni d'un kanban de retrait (circulant entre les poste i et $i - 1$). Ce kanban de retrait est retiré du conteneur et un kanban de production est placé sur ce même conteneur. Le kanban de retrait est renvoyé au $(i - 1)^e$ poste, pour y initier la production. Tout ceci forme une chaîne de production cyclique. Si aucun retrait n'est requis par le poste suivant, le poste précédent ne va rien produire et, ainsi, aucun article excessif ne sera produit. C'est pourquoi, en contrôlant le nombre de kanbans (conteneurs) circulant dans un système *JIT*, une *non-stock-production (NSP)* peut être, en théorie, atteinte. Dans les deux systèmes, une courte distance entre postes successifs est requise; ainsi la circulation des kanbans se fait-elle rapidement. Par contre, si dans un système à un kanban, la synchronisation de la production et du mouvement des conteneurs est nécessaire, ce n'est pas obligatoirement le cas pour un système à deux kanbans. Le kanban accomplit donc les fonctions suivantes :

- visibilité: puisque le kanban et le conteneur sont liés, l'information et les composantes circulent ensemble.
- contrôle de production: il indique quand et quelle quantité de chaque type de composantes doit être produite.
- gestion du stock: le nombre de kanbans mesure la taille du stock et, donc, contrôler le nombre de kanbans équivaut à contrôler le niveau de l'inventaire.

2.2 Principaux critères décisionnels

Le but principal d'un système de production *JIT* étant de réduire autant que possible le degré de l'inventaire du *WIP* et celui d'une gestion de production, de réduire l'espérance des coûts, il semble raisonnable de prendre, pour critères décisionnels, des valeurs qui s'y rapportent. On peut donc citer, par exemple, les différents critères suivants :

- le niveau moyen de l'inventaire du *WIP*,
- le niveau moyen de l'inventaire du *WIP*, sous contrainte que le taux de pénurie ne dépasse pas une valeur donnée,

2.2. PRINCIPAUX CRITÈRES DÉCISIONNELS

- le nombre moyen de ruptures de stock,
- l'espérance du coût total,
- le coût moyen de pénurie,
- le coût moyen de stockage,
- le temps d'attente moyen d'un commande en attente,
- le rapport entre le nombre de commandes non encore honorées et le nombre total de commandes.

Chapitre 3

Modèles mathématiques existants

On dénombre, dans la littérature, une multitude d'heuristiques. Cela s'explique certainement par la complexité d'un tel problème et les difficultés à trouver une heuristique le moins spécifique possible et qui puisse donc s'appliquer à de nombreux cas. Même si cette liste n'est pas exhaustive, voici les principaux modèles :

- Monden,
- Kimura et Terada,
- Bitran et Chang,
- Philipoom, Rees, Taylor et Huang,
- Wang et Wang,
- Bard et Golany.

3.1 Le modèle de Monden (1981) (Formule de Toyota)

Ce modèle est celui que l'on retrouve le plus fréquemment ; cela s'explique par le fait qu'il a été utilisé lors du développement du système Kanban. Monden a établi une liste de règles à respecter, si l'on veut atteindre les buts d'un système de production *JIT*. L'une de ces règles est que le nombre de kanbans entre deux postes de travail doit être minimiser, car il représente le

3.2. LE MODÈLE DE KIMURA ET TERADA (1981)

degré maximal de l'inventaire. Le modèle, relativement simple à comprendre, nous fournit le nombre de kanbans optimal :

$$K = \frac{D * T * (1 + s)}{C}$$

Où:

D : la demande par unité de temps,

T : le temps de production d'un kanban,

C : la contenance d'un kanban,

s : un facteur de marge.

Malheureusement, ce modèle ne tient pas compte des coûts (stockage, pénurie,...) et considère que la demande est constante. Ces faiblesses justifient, à elles seules, le développement de nombreux autres modèles, tels ceux qui suivent ci-dessous.

3.2 Le modèle de Kimura et Terada (1981)

Ils sont les premiers à modéliser un système kanban et leur travail a ainsi servi de références pour de nombreux autres auteurs. Ils supposent que la quantité ordonnée et celle produite sont multiples de la quantité que représente une carte kanban. Ils font également l'hypothèse qu'il n'y a pas de décalage entre la suppression d'une carte et le lancement de la production et que la production au cours d'une période est disponible, dès la prochaine période, pour le poste de travail suivant. Pour cela, ils définissent les notions que voici :

C^n : la capacité de production du n^e poste ;

O_t^n : la quantité à produire ordonnée au n^e poste, durant la t^e période ;

P_t^n : la quantité en train d'être produite au n^e poste, durant la t^e période ;

I_t^n : l'inventaire des composants terminés au n^e poste, à la fin de la t^e période.

Ceci étant fait, ils fournissent leur modélisation sous la forme de trois relations :

$$\begin{cases} O_t^n = P_t^{n-1} + O_{t-1}^n - P_{t-1}^n \\ P_t^n = \min(O_t^n, C^n, I_{t-1}^{n+1} + P_{t-1}^{n+1}) \\ I_t^n = I_{t-1}^n + P_{t-1}^n - P_t^{n-1} \end{cases}$$

La première signifie que la quantité à produire ordonnée au n^e poste est déterminée par les quantités exigées au $(n-1)^e$ poste (i.e. le suivant), auxquelles il faut ajouter les ordres en retard accumulés au cours des périodes précédentes. La deuxième de ces relations signifie que la production est limitée par la quantité à produire ordonnée, la capacité de production du poste et l'inventaire disponible au $(n+1)^e$ poste (i.e. le précédent). La dernière, enfin, représente la balance de l'inventaire au n^e poste. Kimura et Terada n'ont pas directement créé un modèle d'optimisation pour la conduite du système décrit, mais l'ont simulée afin d'étudier l'amplification au n^e poste des fluctuations de la demande au dernier poste. Ce qui leur a permis de conclure qu'il est souhaitable de réduire, au tant que possible, le nombre d'unités représentées par un kanban (autrement dit, sa contenance, sa taille).

3.3 Le modèle de Bitran et Chang (1987)

Ils étendent le travail de Kimura et Terada et offrent un modèle mathématique pour un système kanban pour une chaîne de production à plusieurs postes. Ce modèle a pour but de faciliter la détermination du nombre de kanbans à utiliser à chaque poste et ainsi de contrôler le niveau de l'inventaire. La version présentée ici est une version linéaire, dans laquelle on suppose que les conteneurs remplis partiellement ne sont pas utilisés et que seulement une unité du poste précédent est demandée au poste suivant. Si, comme dans la figure 3.1, P_n représente les points de production et I_n , les points où le *WIP* est inventorié, alors ils supposent également que les kanbans libérés en I_n au cours de la période t sont disponibles et prêts à servir pour la production en P_n , au cours de la période $t+1$. Les auteurs ne considèrent pas les temps de *setup*.

Ils introduisent la notation suivante :

n : étape au cours de laquelle la n^e opération est effectuée ($n=1,2,\dots,N$)

t : la période ($t=1,2,\dots,T$) ;

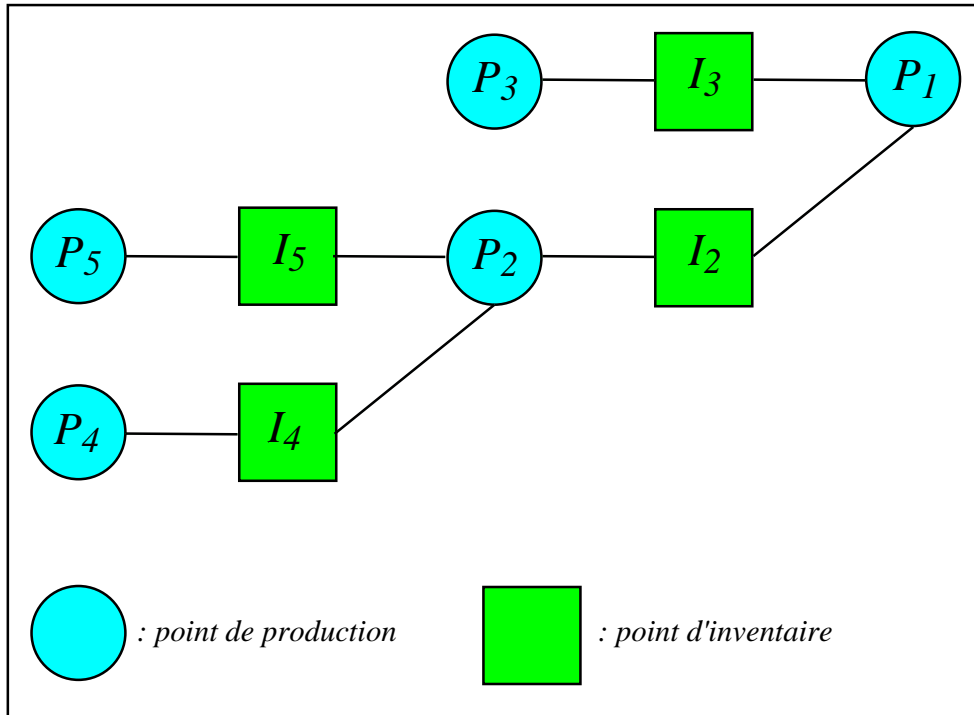


FIG. 3.1: *Points de production et d'inventaire: Bitran et Chang(1987)*

$X_{n,t}$: nombre de kanbans libérés qui déclenchent la production en P_n , au cours de la t^e période;

$Y_{n,t}$: nombre de kanbans libérés en I_n au cours de la t^e période;

$U_{n,t}$: nombre de kanbans libérés et encore inutilisés en P_n , à la fin de la t^e période;

$V_{n,t}$: nombre de conteneurs disponibles en I_n , à la fin de la t^e période;

Q_n : quantité requise de matières premières, déterminée par la quantité commandée et le coût;

$\beta_{n,t}$: capacité de production en terme de conteneurs pleins en P_n , au cours de la t^e période;

α : le nombre de composants dans un conteneur plein;

$s(n)$: étape suivant immédiatement la n^e étape;

$P(n)$: ensemble des étapes précédant immédiatement la n^e étape;

X_t^1 : exigence totale de production, exprimée comme un nombre de conteneurs pleins à la fin de la dernière étape.

Les contraintes du modèle sont les suivantes :

$$\begin{cases} U_{n,t-1} + Y_{n,t-1} - X_{n,t} - U_{n,t} = 0 \\ V_{n,t-1} + X_{n,t} - Y_{n,t} - V_{n,t} = 0 \\ X_{n,t} = \min \begin{cases} U_{n,t-1} + Y_{n,t-1} \\ \beta_{n,t} \\ (V_{k,t-1} + X_{k,t})_{k \in P(n)} \\ Q_n - \sum_{i=1}^{t-1} X_{n,i} \end{cases} \\ Y_{n,t} = X_{s(n,t)} \end{cases}$$

La première de ces contraintes établit la conservation du flux des kanbans en P_n , tandis que la deuxième établit la conservation du flux des kanbans en I_n . La troisième signifie que la production est limitée par la quantité disponible de kanbans, la capacité de production, la quantité de composants disponibles ainsi que le total requis. La dernière contrainte signifie que le nombre de kanbans libérés est égal à la quantité à produire à l'étape suivante. La fonction objectif est de la forme :

$$\text{Min} \sum_{n=1}^N C_n * (U_{n,0} + V_{n,0}),$$

où C_n représente la valeur monétaire d'un conteneur plein, à la n^e étape. Cette fonction doit être interprétée comme une minimisation de la somme pondérée du nombre de kanbans en circulation, à un temps donné; c'est d'ailleurs une borne supérieure pour le *WIP*. En choisissant de minimiser cela au temps $t = 0$, les auteurs ont fixé le nombre maximal de kanbans qui peuvent circuler dans n'importe quelle période.

Les solutions obtenues permettent de guider les directeurs de production dans le choix du nombre de kanbans requis pour rendre le système opérationnel et de suggérer des programmes de production. Cette approche satisfaisante d'optimisation a servi de point de départ à de nombreux autres chercheurs.

3.4 Le modèle de Philipoom, Rees, Taylor et Huang (1990)

Les auteurs considèrent le cas, où il n'est pas possible de réduire les temps de *setup* de toutes les machines au niveau requis pour pouvoir appliquer les méthodes kanbans déjà existantes. Dans le cas qu'ils considèrent, un

3.4. LE MODÈLE DE PHILIPOOM, REES, TAYLOR ET HUANG (1990)

poste de travail créant l'engorgement gênant contient plus d'une machine et produit plus d'un type de composantes, au cours d'un cycle de production ; les temps de *setup* sont longs en comparaison des temps de production. Cette situation se produit dans de petits ateliers qui utilisent des machines ou des outils dont les buts et utilisations sont très généraux, ou dans de plus grandes usines, où les coûts ne permettent pas l'achat de machines spécifiques. Philipoom, Rees, Taylor et Huang montrent que, dans une usine où un nombre limité d'ateliers requièrent de longs temps de *setup*, l'utilisation des méthodes kanbans standards amènent, généralement, à la création d'engorgement, résultants des trops longs et trop fréquents temps de *setup*. Ils présentent deux modèles. L'un a pour but de minimiser l'inventaire généré à la sortie de l'atelier responsable de l'engorgement et le second, de minimiser la somme des coûts de stockage et des coûts de *setup*. Le premier modèle est, ici, détaillé. Les auteurs utilisent la notation suivante :

t_i : la durée d'un cycle de production pour la i^e machine de l'atelier considéré;

P_j : le temps de traitement d'un conteneur de la j^e composante ;

n : le nombre de composantes qui peuvent être produites durant un cycle, dans l'atelier considéré ;

m : le nombre de machines dans l'atelier responsable de l'engorgement ;

S : le temps de *setup* pour les machines de l'atelier considéré, supposé identique pour toutes les composantes ;

q_{ij} : la taille du lot de la j^e composante en train d'être traité sur la i^e machine ;

Q_j : la quantité totale de la j^e composante produite au cours d'un cycle ;

d_j : la demande par unité de temps de la j^e composante ;

Y_{ij} : une variable indicatrice égale à 1 si et seulement si la j^e composante est produite sur la i^e machine ;

M : une constante(arbitrairement élevée).

Le modèle est donné par :

$$\text{Min}Z = \sum_{j=1}^n Q_j$$

sous les conditions :

$$\begin{aligned}
 (1) \quad & t_i \geq \sum_{j=1}^n (q_{ij} * P_j + Y_{ij} * S) \\
 (2) \quad & \begin{cases} \sum_{t=1}^m Y_{ij} = 1 \\ q_{ij} \leq M * Y_{ij} \end{cases} \\
 (3) \quad & \begin{cases} Q_j \leq d_j + (1 - Y_{ij}) * M \\ Q_j \geq d_j - (1 - Y_{ij}) * M \\ Q_j = \sum_{i=1}^n q_{ij} \end{cases}
 \end{aligned}$$

La première des conditions s'assure que la somme du temps de production et du temps de *setup* est inférieure ou égale à la durée d'un cycle. La deuxième s'assure que le j^e article est assigné à une seule machine et la dernière, que la production totale est exactement égale à la demande. La solution, en termes de q_{ij} et de t_j , détermine les quantités à produire et la durée des cycles de production pour chaque produit et chaque machine dans l'atelier responsable de l'engorgement.

3.5 Le modèle de Wang et Wang (1990)

L'objectif de ce modèle est d'appliquer l'approche d'un *processus de Markov* pour déterminer le nombre de kanbans, dans trois configurations de production différentes :

OSTOS : *one station to one station* ;

MSTOS : *multiple stations to one station* ;

MSTMS : *multiple stations to multiple stations*.

Comme la deuxième (respectivement la troisième) est une généralisation de la première (respectivement de la deuxième), seule la première configuration est ici détaillée. Dans ce modèle, Wang et Wang considèrent le nombre de conteneurs pleins entre deux postes de travail comme un processus stochastique. Ainsi le taux d'arrivée du *processus de Markov* est donné par le taux de production du poste de travail précédent et celui de départ, par le taux de consommation du poste de travail suivant. La raison de ce dernier choix est qu'il reflète les caractéristiques d'un *pull-system* ; en effet, dans un *pull-system*, le taux de demande de produits finis par les clients est considéré, au poste de travail en amont, comme le taux de demande local de composantes ou de *WIP*. Les trois principales hypothèses du modèle sont :

- le temps de production et celui de demande sont exponentiellement distribués ;

3.5. LE MODÈLE DE WANG ET WANG (1990)

- le temps de transport entre deux postes de travail est négligeable ;
- le nombre de composants dans un conteneur est constant.

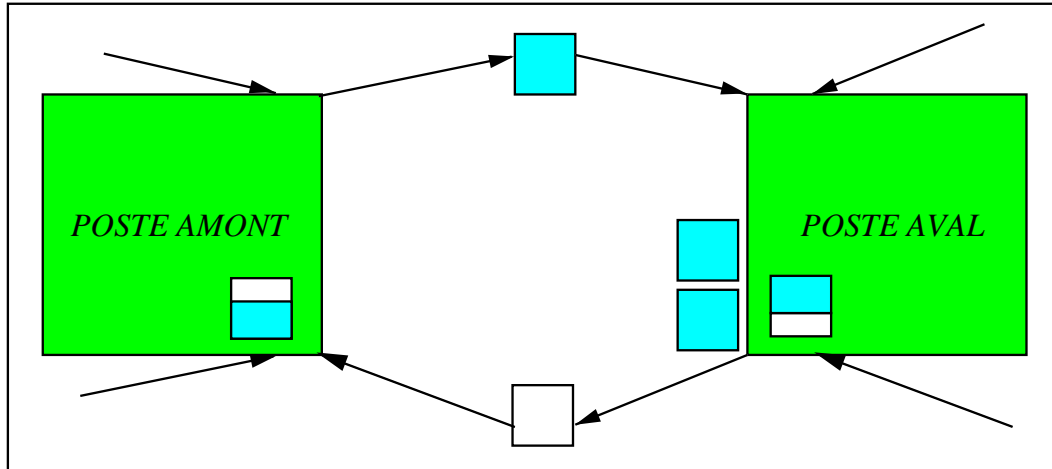


FIG. 3.2: Configuration OSTOS: Wang et Wang(1990)

Pour établir leur modèle, les auteurs utilisent la notation suivante :

E : l'espace des états déterminés par le nombre de conteneurs pleins envoyés au poste de travail suivant ;

K : le nombre de kanbans circulant entre les postes de travail ;

CP : le taux de production du poste de travail amont (exponentiellement distribué) ;

D : le taux de demande du poste de travail aval (exponentiellement distribué) ;

Q_K : la matrice de transition du *processus de Markov* entre les postes de travail, étant donné K kanbans ;

$\pi_K(n)$: la distribution stationnaire du n^e état du *processus de Markov*, avec K kanbans ;

C_n : le facteur de coût total pour être dans le n^e état, par unité de temps ;

S : le coût de pénurie par conteneur, par unité de temps ;

H : le coût de stockage par conteneur, par unité de temps.

Si l'on suppose qu'il y a K kanbans entre les deux postes de travail, le système peut être décrit comme un *processus de Markov* avec $NS = K + 1$ états ($E = \{0, 1, 2, \dots, K\}$). Il faut ensuite définir la matrice de transition Q_K ; trois situations sont à considérer :

1. Pour l'état 0 (qui signifie que tous les conteneurs sont vides) : on suppose qu'il n'y a plus de production au poste de travail aval ; seul le poste de travail amont produit les composantes requises aussi vite que possible. En d'autres termes, il n'y a qu'un taux d'arrivée et le seul état atteignable est l'état 1 ;
2. Pour l'état K (qui signifie que tous les conteneurs sont pleins) : il n'y a plus de production au poste de travail amont ; seul le poste de travail aval consomme les composantes. En d'autres termes, il n'y a qu'un taux de départ et le seul état atteignable est l'état $K-1$;
3. Pour les autres états, il y a un taux d'arrivée et un taux de départ. Du n^e état, le $(n - 1)^e$ état ainsi que le $(n + 1)^e$ état peuvent être atteints.

En posant :

$$p_1 = \frac{D}{CP+D} \text{ et}$$

$$p_2 = \frac{CP}{CP+D} = 1 - p_1,$$

on obtient la matrice de transition Q_K suivante :

$$Q_K = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ p_1 & 0 & p_2 & 0 & \cdots & 0 \\ 0 & p_1 & 0 & p_2 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & p_1 & 0 & p_2 \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

La distribution stationnaire se trouve en effectuant les étapes suivantes :

1. Trouver l'unique solution de l'équation $v = v*Q_K$, où $v = (v_0, v_1, \dots, v_K)$.
2. Le nombre moyen de visites du i^e état entre deux visites d'un certain état l vaut v_i/v_l et le temps moyen passé dans le l^e état vaut $1/\lambda_l$, où $\lambda = (\lambda_0, \dots, \lambda_K)$ est le vecteur des paramètres des distributions exponentielles des temps de séjour dans les différents états.

3. La durée moyenne d'un cycle (temps écoulé entre deux visites successives du l^e état) est donné par:

$$\frac{1}{\lambda(l)} + \frac{1}{v(l)} * \sum_{i \neq l} \frac{v(i)}{\lambda(l)}$$

4. Le temps moyen passé dans le l^e état est donné par :

$$\pi_K(l) = \frac{\frac{v(l)}{\lambda(l)}}{\sum_{i=0}^K \frac{v(i)}{\lambda(i)}}$$

Le coût total moyen TC_K par unité de temps, pour un système à K kanbans est fourni par l'équation suivante:

$$E(TC_K) = \sum_{i=0}^K \pi_K(i) * C_i$$

où les C_i se trouvent comme suit :

1. Pour l'état 0: $C_0 = S$;
2. Pour l'état K : $C_K = H * K$, en supposant qu'il n'y a pas de coût pour l'absence de travail au poste amont ;
3. Pour le n^e état: $C_n = H * n$.

Grâce au coût total moyen, les auteurs déterminent le nombre de kanbans optimal K^* , de manière graphique en représentant $E(TC_K)$ en fonction de K .

3.6 Le modèle de Bard et Golany (1991)

Bard et Golany examinent le problème consistant à déterminer le nombre de kanbans optimal dans un système de production multistage, multiproduit. Comme le montre la figure 3.3, ils décrivent l'exemple d'une usine d'assemblage de circuits imprimés, où cinq postes de travail produisent trois articles finis différents. Un poste de travail est en principe capable d'entreprendre plus d'une opération et, donc, de générer plus d'un article non fini. L'objectif du modèle est de déterminer le nombre de kanbans à utiliser à chaque poste de travail de façon à minimiser le coût total, incluant les coûts *setup*, les

coûts de stockage et les coûts de pénurie, au cours d'un intervalle de temps donné, connu. Les notations définies par les auteurs sont les suivantes :

– Ensembles et indices :

I : l'ensemble des composantes ou articles : $i = 1, \dots, N_i$;

E : l'ensemble des articles finis ($I \supset E$) ;

W : l'ensemble des postes de travail : $k = 1, \dots, N_k$;

T : l'ensemble des périodes (de temps) : $t = 1, \dots, T_t$;

$I(k)$: l'ensemble des articles produits au k^e poste de travail ;

$S(i)$: l'ensemble des successeurs immédiats de i ;

$P(i)$: l'ensemble des prédécesseurs de i .

– Paramètres fournis :

d_{it} : demande espérée pour le i^e article, durant la t^e période ;

c_{kt} : capacité de production au k^e poste de travail, durant la t^e période ;

h_i : coût de stockage par conteneur, par période, pour le i^e article ;

s_i : coût de pénurie par conteneur, par période, pour le i^e article ;

f_i : coût de *setup* pour le i^e article ;

b_i : temps de production d'un lot du i^e article ;

σ_i : temps de *setup* pour un lot du i^e article ;

g_i : délai d'exécution/réalisation pour le i^e article ;

a_{ij} : rapport entre le nombre de conteneurs du i^e article et le nombre de conteneurs du j^e article ;

M : une constante (arbitrairement élevée).

– Variables de décisions :

X_{it} : nombre de conteneurs du i^e article produits durant la t^e période ;

Z_{it} : variable indicatrice égale à 1 si et seulement si le i^e article est produit durant la t^e période ;

I_{it} : nombre de conteneurs du i^e article, à la fin de la t^e période ;

Ψ_i : nombre de conteneurs du i^e article dans le système.

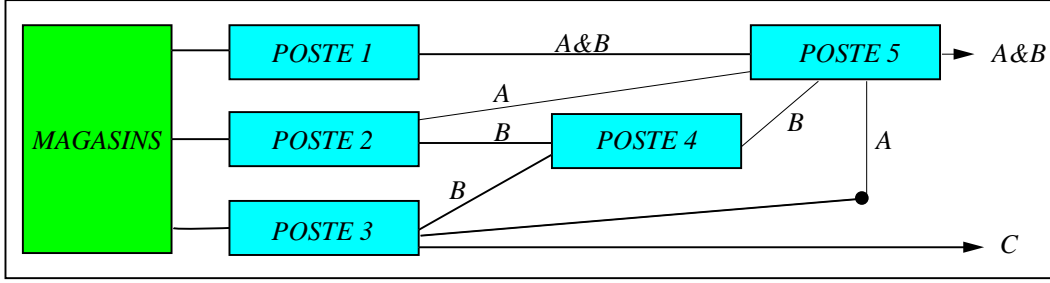


FIG. 3.3: Flux des composantes de A,B,C: Bard et Golany(1991)

Le modèle d'optimisation possède une fonction objectif comportant deux termes principaux ; le premier somme les coûts de stockage et ceux de *setup* pour les articles en cours de production, tandis que le second somme les coûts de stockage, les coûts de pénuries et ceux de *setup* pour les articles finis. Il semble raisonnable d'appliquer les coûts de pénuries uniquement aux articles finis. Ces termes sont sommés sur l'intervalle de temps total. Dans la première contrainte, les auteurs s'assurent simplement d'avoir l'équation de la balance matérielle ; il faut noter qu'excepté pour les articles finis, le terme d_{it} est nul. La seconde contrainte possède trois parties et limite la production d'un article particulier. La première de ces parties stipule que la production d'un article à un poste de travail est limitée par la disponibilité des composantes requises, produites aux postes de travail précédents ; la seconde, que la production est limitée par le nombre de kanbans en circulation au poste de travail considéré ; la dernière de ces parties établit que la production est limitée par la capacité de production du poste de travail considéré. La troisième contrainte s'assure qu'une production ne sera lancée qu'au cas où un *setup* a été précédemment effectué. Le modèle est le suivant :

$$\text{Min} \sum_{t \in T} \left\{ \sum_{i \in I|E} [h_i * I_{it} + f_i * Z_{it}] + \sum_{i \in E} [h_i * \max(I_{it}, 0) + s_i * \max(-I_{it}, 0) + f_i * Z_{it}] \right\}$$

sous les conditions :

$$\begin{aligned} (1) \quad & I_{i,t-1} + X_{i,t-g_i} - I_{jt} - \sum_{j \in S(i)} a_{ij} * X_{jt} = d_{it} & \forall i \in I, t \in T \\ (2) \quad & X_{it} = \min \begin{cases} \min_{j \in P(i)} \max(I_{j,t-1}, 0) / a_{ji} \\ \Psi_i - I_{i,t-1} \\ [c_{kt} - \sum_{\alpha \in I(k)} (b_\alpha * X_{\alpha t} + \sigma_i * Z_{it})] / b_i \end{cases} \\ (3) \quad & X_{it} \leq (c_{k,t-\sigma_i} / b_i) * Z_{it} & \forall i \in I, t \in T \end{aligned}$$

$$X_{it} \geq 0, Z_{it} \in \{0, 1\}, \Psi_i \geq 0, I_{it} \text{ non-restreint } \forall i \in I, t \in T.$$

La fonction objectif est non-convexe. Les auteurs proposent une procédure en trois étapes pour chercher un optimum global. Dans l'exemple proposé, ils ont obtenu un nombre de kanbans oscillant entre 15 et 30, avec une moyenne de 22.3. En utilisant la formule de Toyota, on trouve une moyenne de 13.2 kanbans. Ce modèle semble donc préconiser l'utilisation de grande quantité de kanbans ; cela s'explique par le fait que les coûts sont, ici, explicitement tenus en compte.

3.7 Conclusion

Après ces présentations détaillées, il apparaît utile, pour la synthèse, de regrouper ces différents modèles :

	Monden	Kimura Terada	Bitran Chang
HYPOTHESES	-demande constante -pas de coûts	-pas de temps -de transport -production en série	-multistage -demande=1 -pas de setup
NATURE DU MODELE	-formule -déterministe	-simulation -déterministe	-optimisation -déterministe -linéaire
OBJECTIF	-fournit le nombre de kanbans	-étude des amplifications de fluctuations de la demande	-fournit le nombre de kanbans optimal

3.7. CONCLUSION

	Philipoom Rees Taylor Huang	Wang Wang	Bard Golany
HYPOTHESES	-setup longs -petite usine -machines -multitâches -engorgement	-taille des kanbans=cste -pas de temps de transport temps=v.a. exponetielles -coût: .stockage .pénurie	-multistage -multiproduit -setup -coût: .stockage .pénurie
NATURE DU MODELE	-optimisation -déterministe -linéaire	-optimisation -stochastique -processus de Markov	-optimisation -déterministe -fct objectif non convexe
OBJECTIF	-diminuer: .l'inventaire à . l'engorgement -coûts (stock+setup)	-fournit le nombre de kanbans optimal tq.coût total d'espérance minimal	-fournit le nombre de kanbans optimal tq.coût total d'espérance minimal

Tous ces modèles cherchent à déterminer les meilleures valeurs pour différents paramètres, comme par exemple le nombre de kanbans à utiliser. La complexité mathématique des formulations démontre que la solution pour de grands problèmes n'est pas du tout aisée. Cependant, la spécification formelle de ces différents modèles d'optimisation encourage le développement, pour d'autres cas particuliers, d'heuristiques plus ou moins simples et des tentatives de généralisation de modèles. L'utilisation de fonctions objectifs basées sur les coûts soulève d'ailleurs d'intéressantes questions. Même s'il ne fait aucun doute que les décisions prises, dans la gestion de production, cherchent, plus que jamais, à réduire ces coûts, il existe tout de même un danger que ces modèles recommandent des choix de paramètres qui ne fassent pas diminuer la taille de l'inventaire du *WIP*. Ce qui reste tout de même l'idée de base des systèmes de production *JIT*.

Chapitre 4

Développement d'un nouveau modèle

Pour développer ce nouveau modèle, je me base sur l'heuristique de *Wang et Wang* (cf. section 3.5). Le but est également d'appliquer l'approche d'un *processus de Markov* à la détermination du système de production optimal. La distinction des trois situations *OSTOS*, *MSTOS* et *MSTMS* est également conservée, mais, par manque de temps, seul le premier cas de figure a été, pour l'instant, développé. Je considère la situation suivante: un consommateur et son fournisseur décident de collaborer par kanbans interposés. Là où le modèle devient novateur et s'éloigne encore des précédentes heuristiques (cf. chapitre 3), réside dans la postulation de l'existence d'une politique de production chez le fournisseur. C'est-à-dire qu'il ne va plus produire dès réception d'un kanban comme dans le modèle de *Wang et Wang*. Tout cela, parce que le fournisseur connaît également des coûts fixes, tels ceux engendrés par les temps de *setup*. Je suppose que le nombre de conteneurs pleins circulant entre le fournisseur et le consommateur est un processus stochastique. Le taux d'arrivée, égal au taux de production d'un lot de kanbans de taille L , est donné par :

$$T_a = \frac{1}{\sigma + \frac{L}{\mu}}$$

et celui de départ, égal au taux de consommation d'un kanban, par :

$$T_d = \lambda$$

où:

σ = temps de setup moyen ;

L = nombre de kanbans en train d'être produits chez le fournisseur ;

μ = taux de production d'un kanban ;

λ = taux de consommation d'un kanban.

4.1 Politiques de production

Dans le modèle de *Wang et Wang*, l'hypothèse que le fournisseur engage la production dès réception d'un seul kanban n'est pas réaliste et fortement réductrice. Il est inconcevable qu'une entreprise qui subit des coûts de production fixes et des charges d'exploitation, fassent preuve d'autant de serviabilité, sans courir à sa perte. Ceci d'autant plus que ce fournisseur commerce certainement avec plusieurs clients. C'est pourquoi, en général, il décide de lancer sa production en tenant compte des coûts qui en découlent et, par exemple, des priorités qui peuvent exister entre ses différents clients. Je choisis de modéliser cela comme suit. Dorénavant, le fournisseur ne va plus produire dès réception d'un conteneur vide. Il établit deux niveaux de production compris entre les valeurs 1 et le nombre total de kanbans circulant entre les deux protagonistes. Le premier est appelé niveau d'engagement (*NE*) et le second, niveau d'urgence (*NU*) (cf. figure 4.1). Tant qu'il n'a pas, chez lui, une quantité de conteneurs vides égale ou supérieure au niveau d'engagement, il ne lance pas de production ; si, en revanche, cette quantité est égale ou supérieure au niveau d'urgence, il produit avec une probabilité égale à 1. Entre ces deux limites, le lancement d'une production est une variable aléatoire binomiale conditionnée sur le nombre de kanbans vides que possède le fournisseur. C'est-à-dire que, pour un nombre de kanbans compris entre *NE* et (*NU-1*), il lance une production avec une probabilité donnée. Cette démarche, même si elle semble s'éloigner des systèmes *JIT*, se justifie par l'existence de coûts fixes et une volonté de rapprochement à la réalité industrielle et économique. Pour mon modèle, je considère quatre politiques de production différentes. Par commodité et parce que leurs graphes sont évocateurs (cf. figure 4.1), je les ai baptisées :

- politique *linéaire* ;
- politique *exponentielle* ;
- politique *logarithmique* ;
- politique *constante*.

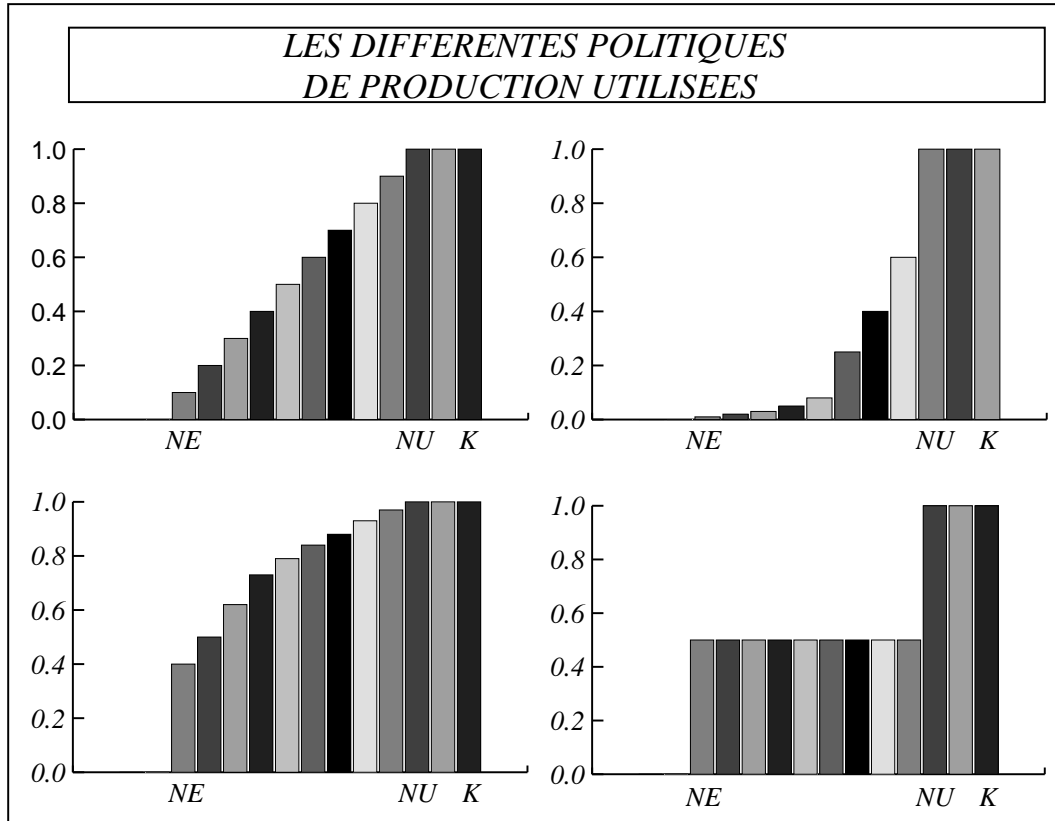


FIG. 4.1: A chaque quantité de kanbans est associée une probabilité de production

4.2 Hypothèses et notations

Les principales hypothèses du modèle sont donc les suivantes :

- le temps de production d'un lot de kanbans de taille L est une variable aléatoire de loi $\varepsilon(\sigma + \frac{L}{\mu})$;
- le temps de consommation d'un kanban est une variable aléatoire de loi $\varepsilon(\lambda)$;
- le temps de *setup* est pris en compte ;
- le temps de transport entre les postes de travail est négligeable ;
- il existe une politique de production définie par deux bornes (NE et NU) ; le lancement d'une production est une variable aléatoire binomial

4.2. HYPOTHÈSES ET NOTATIONS

conditionnée sur le nombre de kanbans ;

– la taille des conteneurs est fixe.

Pour établir mon modèle, j'utilise la notation suivante :

K : nombre total de kanbans circulant dans le système ;

NE : niveau d'engagement ;

NU : niveau d'urgence ;

λ : taux de consommation d'un kanban ;

μ : taux de production d'un kanban ;

σ : temps moyen de *setup* nécessaire au lancement de la production d'un lot ;

L : taille du lot en train d'être produit chez le fournisseur ;

E : ensembles des états du système ;

NS : cardinalité de E ;

A : matrice génératrice du *processus de Markov* ;

Π^* : distribution stationnaire du système ;

$max_{penurie}$: taux maximal de pénurie admis.

4.3 Etats et matrice génératrice

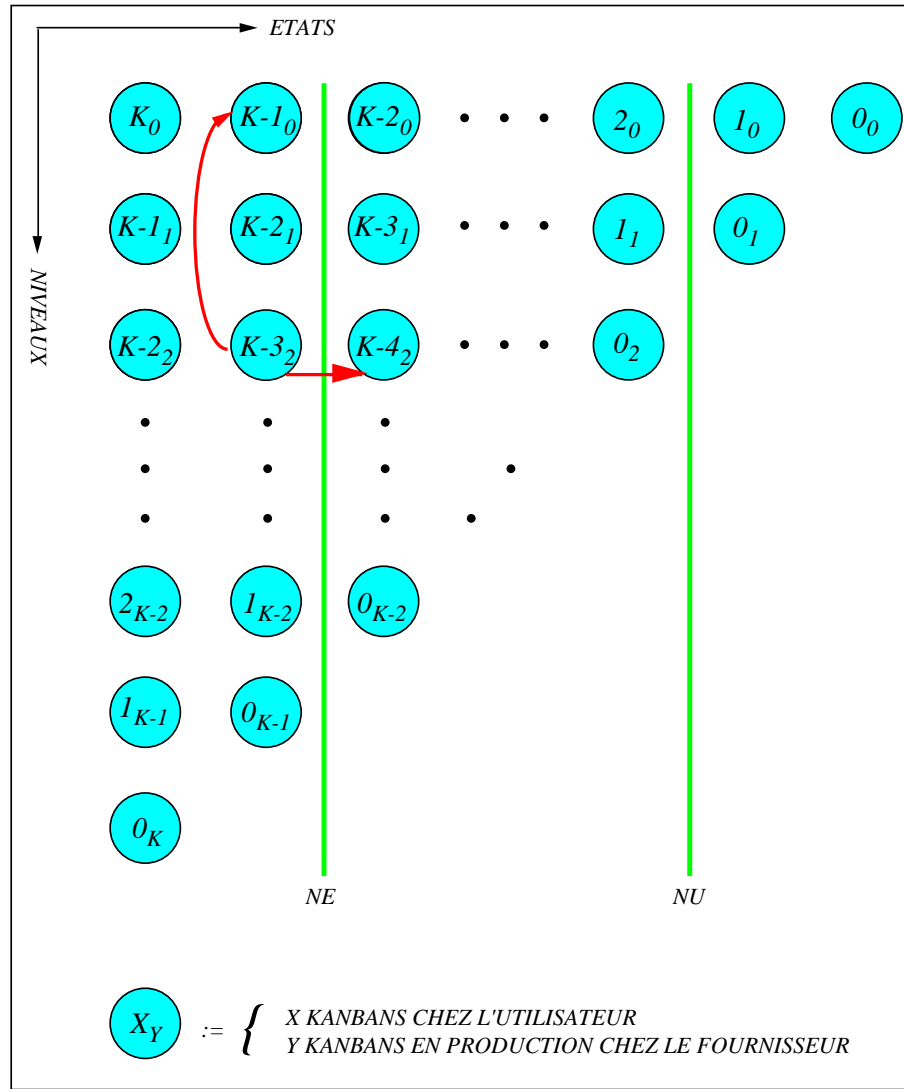


FIG. 4.2: Ensemble des états du modèle OSTOS

L'ensemble E est défini comme suit (cf. figure 4.2). Chaque état (X, Y) est uniquement déterminé par le nombre de kanbans qui se trouvent en stock chez l'utilisateur (X) et par le nombre de kanbans que le fournisseur est en train de remplir (et non pas ceux qui sont en attente chez lui) (Y). Le nombre d'états distincts est donc facilement calculable. Dans un premier temps, on

4.4. DÉFINITION DES TRANSITIONS

trouve :

$$NS = \frac{(K+1)(K+2)}{2}$$

Cependant, même si la matrice génératrice fournit les transitions possibles à partir de tous les états, certains sont inatteignables et ne seront, par conséquent, jamais visités. (cf. figure 4.3) Par exemple, les états suivants :

$(0,0)$: signifie que le producteur n'a pas relancé de production alors qu'il possède K kanbans chez lui ; or $K \geq NU$, par définition.

$(X, NU+1)$: étant donné que le producteur produit, avec une probabilité égale à un, dès qu'il a chez lui K kanbans, il ne lancera jamais une production d'un lot de taille supérieure à NU .

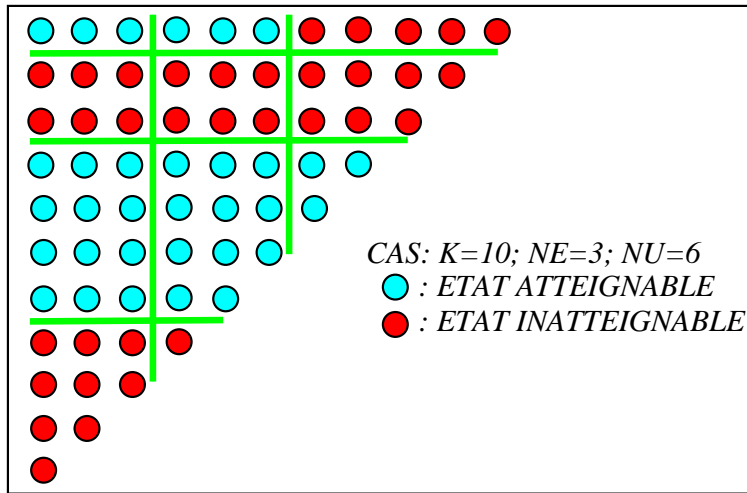


FIG. 4.3: *Etats visités et états non visités*

Donc, finalement, le nombre d'états distincts (atteignables) est donné par :

$$NS = (NU - NE + 1) * K + \frac{(NE - 1) * (NE - 2) - NU * (NU - 3)}{2}$$

4.4 Définition des transitions

Dans le cas général, les transitions sont fixées comme suit. Si l'on se trouve dans l'état (X, Y) , les états $(X-1, Y)$, $(X+Y, 0)$ et $(X+Y, K-X-Y)$ sont atteignables avec une certaine probabilité, dépendant de la politique de production du fournisseur. Il est évident qu'il faut encore distinguer les cas $X = 0$

ou $Y = 0$. L'ensemble des transitions possibles est donné dans la figure 4.4.

La matrice génératrice A contient les taux de transitions entre les différents états et ses termes diagonaux (égaux aux opposés des sommes des termes de chaque lignes) nous fournissent le taux de passage en dehors de chaque état. Etant donné que ce processus est ergodique, on sait que la distribution stationnaire Π^* se trouve en résolvant le système suivant :

$$\begin{cases} \Pi * A = 0 \\ \Pi * Id = 1 \end{cases}$$

Une fois la distribution stationnaire trouvée, n'importe quelle espérance de coût peut être aisément calculée, de la même façon que dans le modèle de *Wang et Wang*. Il suffit pour cela de définir les valeurs des différents coûts associés à chaque état. Pour ma part, je m'intéresse aux problèmes suivants :

- Etant donné K et une politique de production, trouver le couple (NE, NU) minimisant le niveau du stock, sous la contrainte que le taux de pénurie soit inférieur à $max_{penurie}$.
- Etant donnée une politique de production, trouver K^* minimisant le niveau du stock, sous la contrainte que le taux de pénurie soit inférieur à $max_{penurie}$. Dans cette situation, NE et NU sont fonctions de K . Par exemple, $NE = \lceil K/3 \rceil$ et $NU = \lceil 2 * K/3 \rceil$

Ces différents problèmes d'optimisation ont été codés (cf. appendice A) de manière à obtenir une résolution à l'aide du logiciel *Matlab 5.3.0*.

4.4. DÉFINITION DES TRANSITIONS

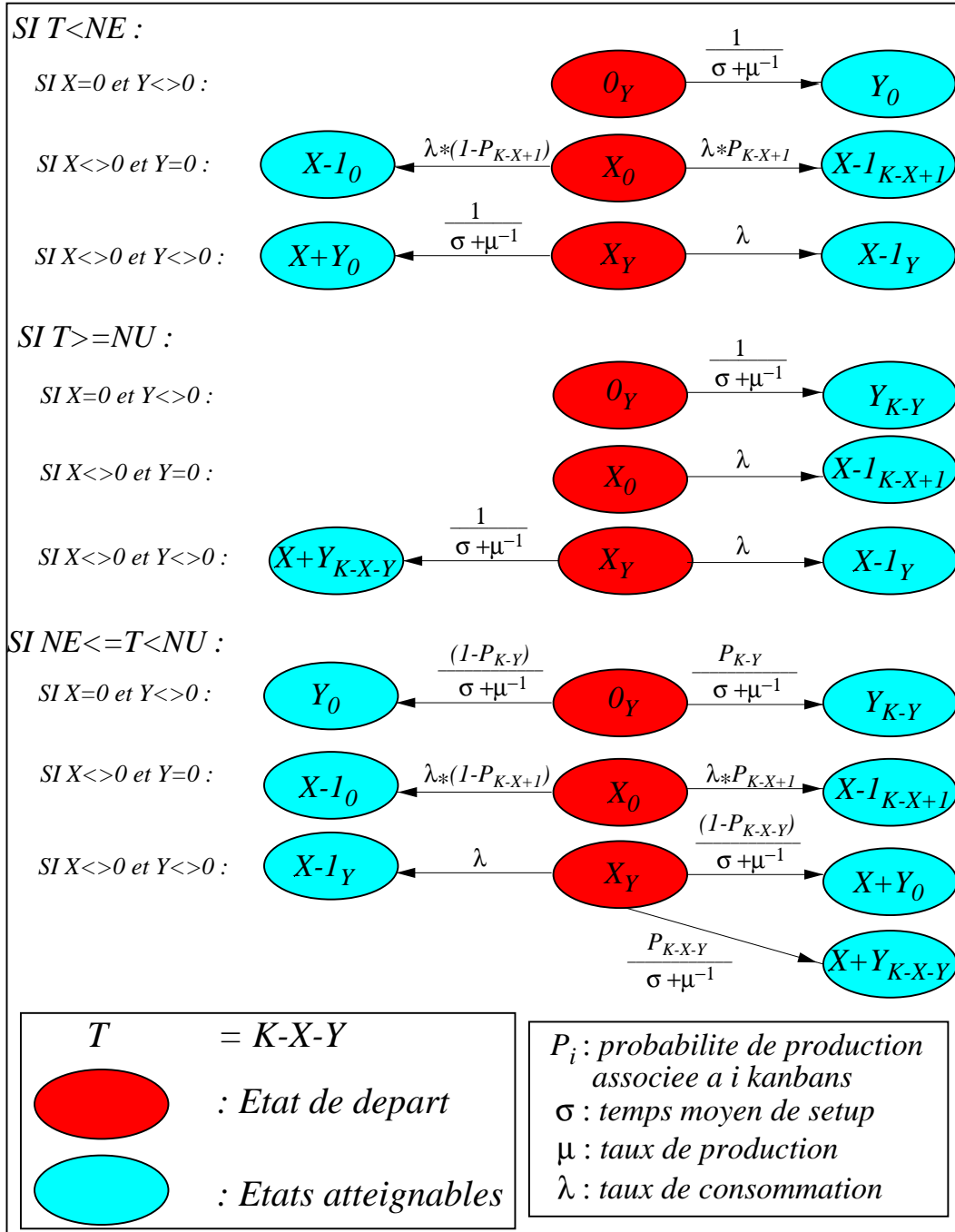


FIG. 4.4: Transitions possibles et taux de transitions dans le modèle OSTOS

Chapitre 5

Résultats et analyse du modèle

5.1 Comparaison avec le modèle de Wang et Wang

Puisque le modèle développé a pour base celui de *Wang et Wang*, il paraît intéressant de les confronter. Cela permet de savoir, si les hypothèses (existence d'une politique de production) ainsi que les contraintes (existence d'un temps de *setup*) supplémentaires sont raisonnables et ne rendent pas le modèle caduque. Dans ce but, je représente les courbes du niveau de stock moyen et du taux de pénuries moyen en fonction du nombre de kanbans K , pour chaque politique de production et pour le modèle de *Wang et Wang*. Etant donné que dans cette dernière heuristique, le temps de *setup* est ignoré, il est tout d'abord intéressant de les comparer pour un temps de *setup* quasiment nul (cf figure 5.1, à gauche). Pour pouvoir comparer, il faut également faire subir cette contrainte au modèle de *Wang et Wang*. C'est donc déjà un modèle quelque peu modifié que l'on utilise ; toutefois, cette modification est nécessaire pour placer les différents modèles sur un pied d'égalité. On observe que :

- le niveau de stock moyen est une fonction linéaire du nombre de kanbans ;
- le taux de pénuries moyen décroît de façon exponentielle, quand croît le nombre de kanbans.

Et cela, pour le modèle de *Wang et Wang* ou pour n'importe quelle politique de production. Les principales propriétés d'un système Kanban sont donc effectivement conservées. Contrôler le nombre de kanbans équivaut toujours à contrôler le niveau de l'inventaire. On remarque également, que le nouveau

5.1. COMPARAISON AVEC LE MODÈLE DE WANG ET WANG

modèle possède un niveau de stock moyen nettement inférieur à celui de *Wang et Wang*. Toutefois, ceci se paie d'une augmentation du taux de pénurie. En effet, plus le niveau de stock est bas, plus le taux de pénuries est mauvais. Remarquons encore que, si l'on augmente le temps de *setup*, le modèle de *Wang et Wang* perd de son excellence et que la politique logarithmique fait preuve d'un comportement intéressant (cf figure 5.1, à droite). si l'on continue d'augmenter (cf figure 5.2), le choix d'une politique n'a plus vraiment de sens, car les cinq modèles convergent. Il apparaît donc fondamental de considérer le temps de *setup*, vu son influence. Les résultats obtenus pour un grand temps de *setup* s'expliquent par la construction même du modèle : son absence de mémoire due à l'utilisation de variables aléatoires exponentielles. Une des fâcheuses répercussions est que le taux de pénurie est certainement surestimer. D'où l'intérêt d'un perfectionnement de ce modèle (cf. chapitre 6).

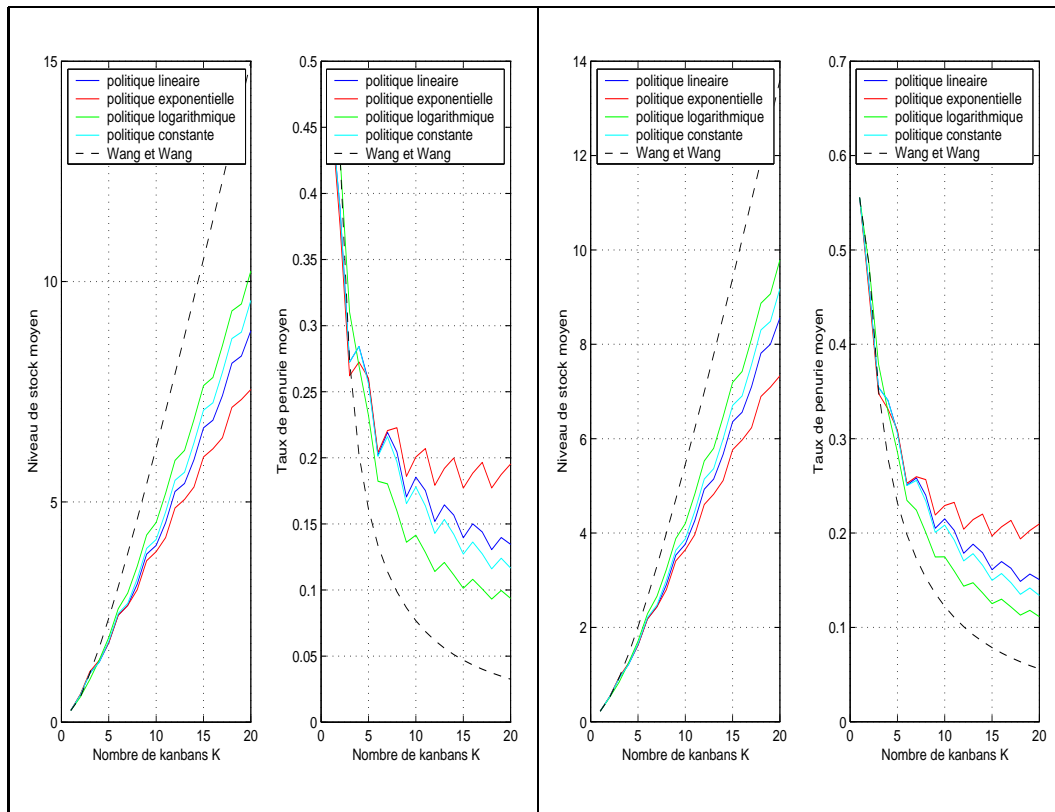


FIG. 5.1: Niveau de stock et taux de pénuries moyens ($K=20, \lambda = 5, \mu = 8$), pour $\sigma = .005$ ou $\sigma = .125$.

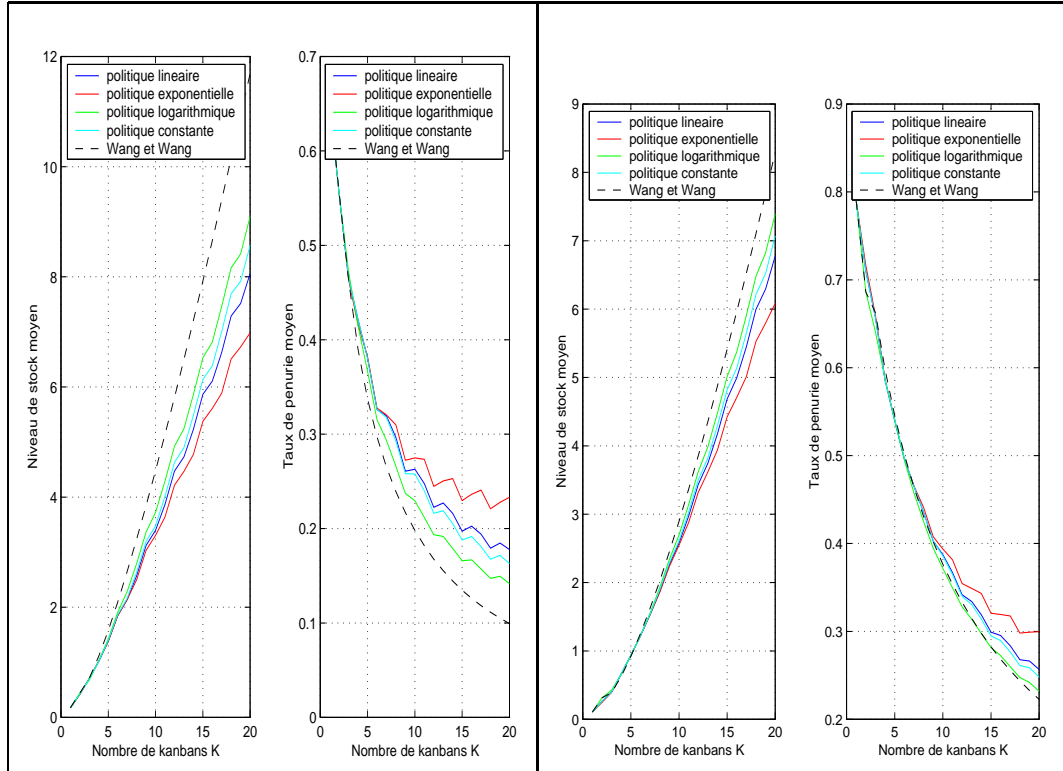


FIG. 5.2: Niveau de stock et taux de pénuries moyens ($K=20$, $\lambda = 5$, $\mu = 8$), pour $\sigma = .25$ ou $\sigma = .625$.

5.2 Influence du couple (NE,NU) dans le graphique

En observant les figures 5.1 et 5.2, on remarque que la courbe du taux de pénuries, quelque soit la politique, présentent un certain nombre de pics et de creux régulièrement espacés. Cela vient du fait que pour ces graphiques, les niveaux d'engagement et d'urgence sont arbitrairement fixés. En effet, puisque seul K doit varier, je décide de les poser égaux aux valeurs entières supérieures d'une fraction de K . Les pics sont le résultat du choix de ces deux fractions. Cependant, comme le montre la figure 5.3, leur influence sur l'allure globale de la courbe est faible.

5.3. COMPARAISON DES POLITIQUES

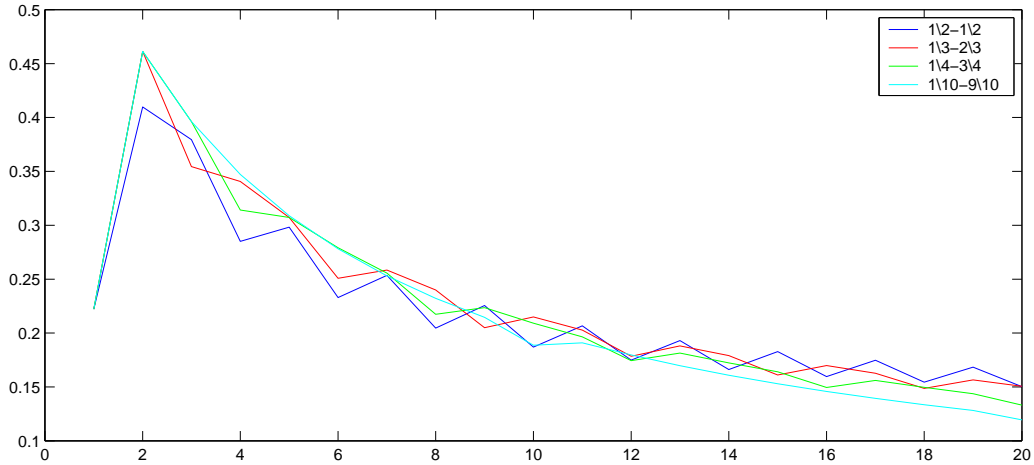


FIG. 5.3: Comparaison des taux de pénuries moyens pour différents choix de NE et de NU

$K = 20$ $\lambda = 5, \mu = 8$ $max_{penurie} = .15$	politique	lin.	exp.	log.	cst.
	<i>Stock</i>	8.52	8.56	8.65	8.63
$\sigma = .125$	<i>NE</i>	8	8	9	9
	<i>NU</i>	12	11	20	10
	<i>Stock</i>	8.82	8.82	8.73	8.81
$\sigma = .25$	<i>NE</i>	8	8	8	8
	<i>NU</i>	8	8	11	8
	<i>Stock</i>	9.7	9.69	9.68	9.72
$\sigma = .375$	<i>NE</i>	2	4	4	3
	<i>NU</i>	7	5	7	6

FIG. 5.4: Niveau moyen de stock minimal et (NE, NU) associé pour chaque politique et plusieurs taux maximaux de pénuries

5.3 Comparaison des politiques

En comparant les valeurs du tableau 5.4, ou les graphiques de la figure 5.5, on remarque qu'au niveau qualitatif, les résultats sont sensiblement pareils, quelque soit la politique de production utilisée. Lorsque le temps de *setup* augmente, c'est la politique logarithmique qui est sensiblement la

meilleure. Cela vient étoffer les observations effectuées à partir des figures 5.1 et 5.2. En regardant la figure 5.5, on remarque que la politique exponentielle se distingue en proposant beaucoup plus de couples (NE, NU) tels que le niveau de stock moyen soit minimal ; cependant, si l'on tient compte du taux de pénuries moyen, on voit que le taux associé à ces couples est très élevé. Ils sont donc sans réel intérêt. D'ailleurs, ce qu'il ressort de ces différentes figures est que, même si qualitativement les quatre politiques sont proches, la politique exponentielle est à oublier. En effet, au niveau quantitatif cette fois-ci, elle est de très loin la plus mauvaise politique. De plus, le fait que la politique logarithmique soit la meilleure est quelque peu rassurant. Même s'il ne s'agit que d'une modélisation simpliste de prises de décisions a priori complexes, c'est bien celle qui semble se rapprocher le plus de la réalité et d'une véritable politique de production.

5.3. COMPARAISON DES POLITIQUES

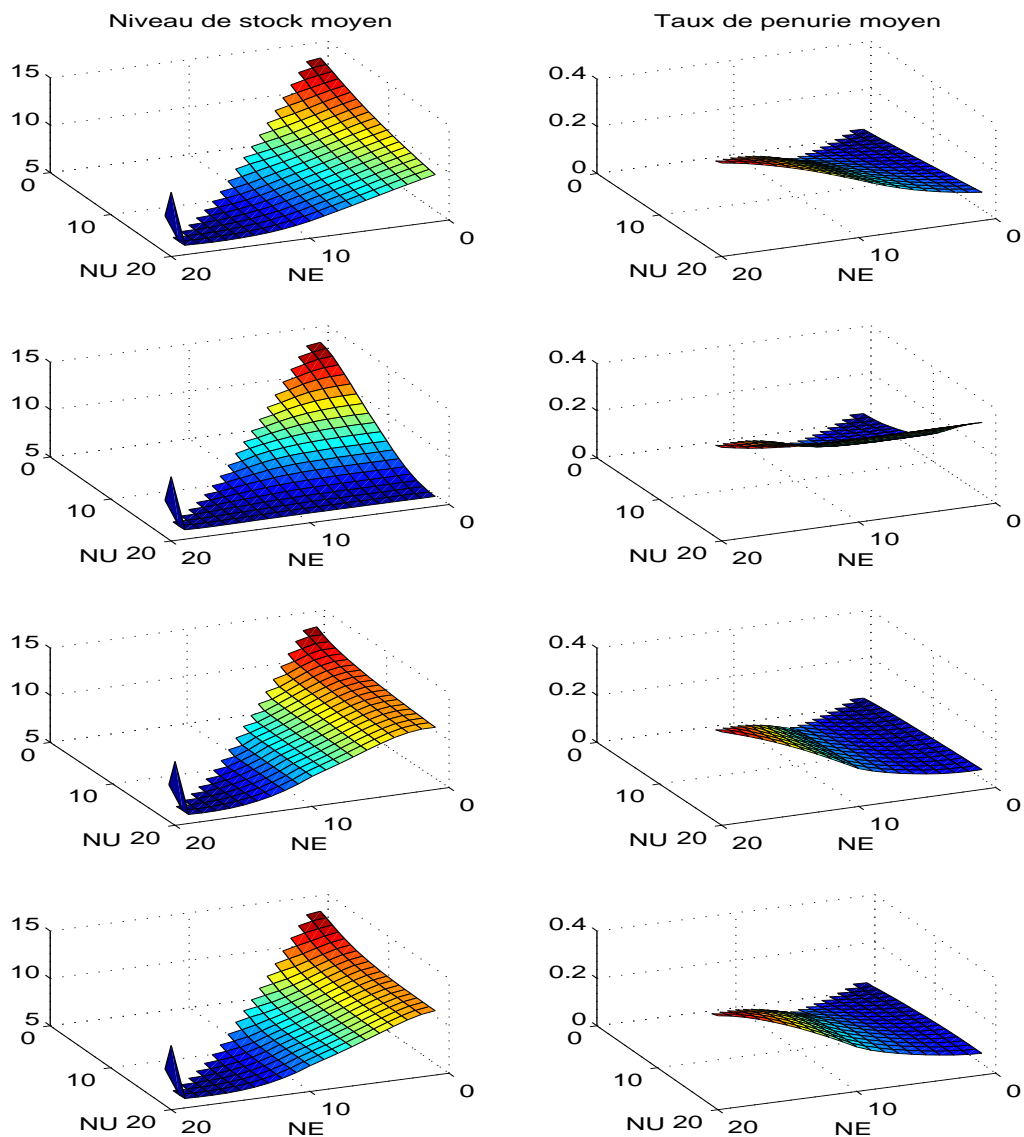


FIG. 5.5: Niveau de stock et taux de pénuries moyens en fonction de NE et NU ($K = 20$) pour les politiques: linéaire, exponentielle, logarithmique et constante

5.4 Evolution de (NE, NU) optimaux en fonction de K

Dans la figure 5.6 s'affichent les valeurs du couple (NE, NU) optimal (i.e. minimisant le niveau de stock sous contrainte d'un taux de pénuries maximal égal à .10) en fonction de K , le nombre total de kanbans circulant dans le système. Jusqu'à la valeur $K = 11$, les courbes sont confondues avec celle du niveau de stock moyen associé; cela vient du fait qu'il n'existe pas de valeurs (NE, NU) tels que le taux maximal de pénuries ne soit pas dépassé. Ensuite, il est intéressant d'observer qu'il existe bien une structure, que les paramètres du modèle forcent un choix de (NE, NU) qui n'est apparemment pas le fruit du hasard.

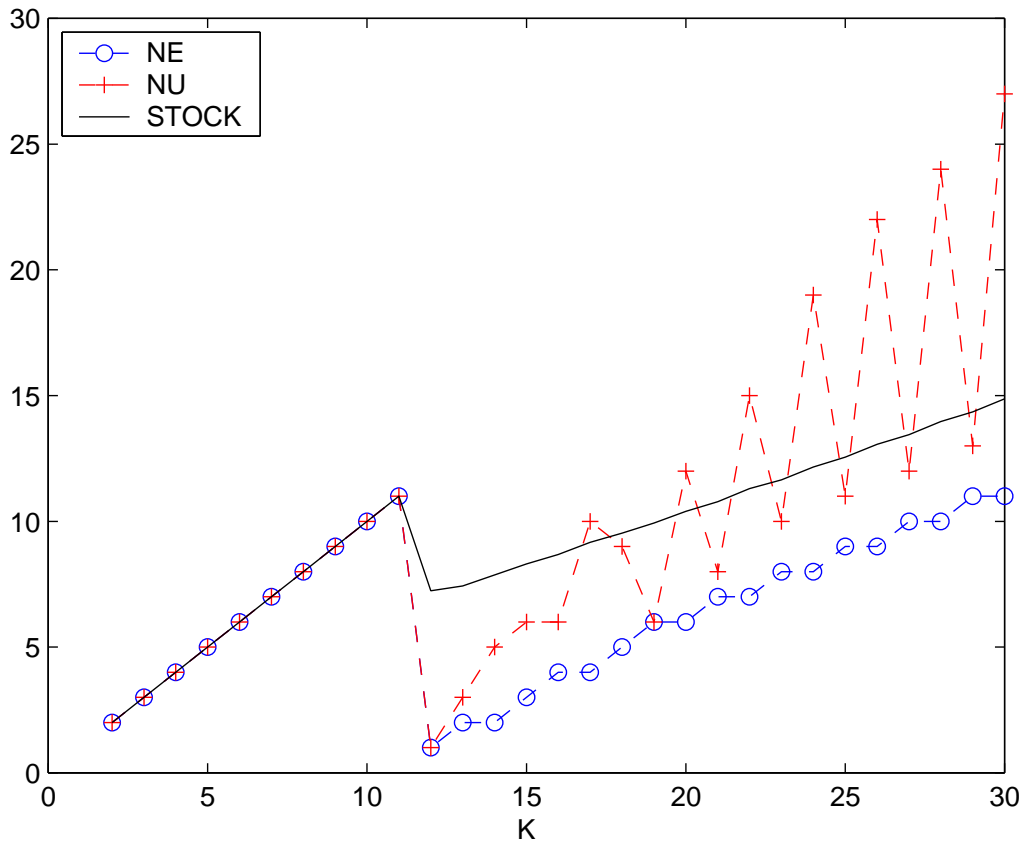


FIG. 5.6: Niveau de stock moyens, NE et NU optimaux, en fonction de K

Chapitre 6

Possibilités d'extension du modèle

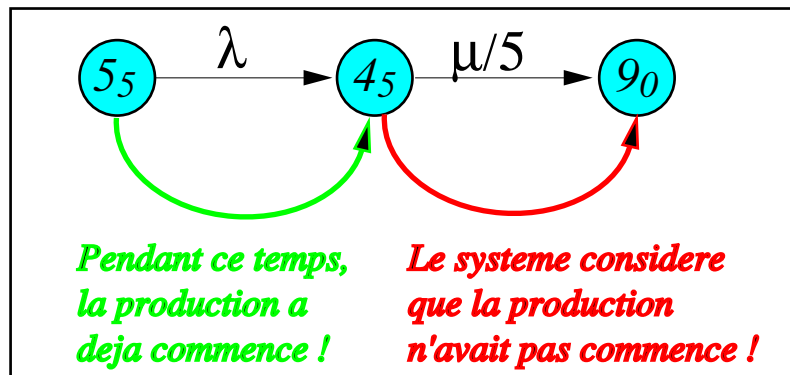


FIG. 6.1: Absence de mémoire du modèle

La prochaine étape a pour but de lutter contre l'absence de mémoire du modèle. Toutefois, même si l'on sait parfaitement quelle en est la source, on conserve le processus markovien comme charpente; c'est-à-dire que les temps de production et de consommation sont toujours des variables aléatoires exponentielles. C'est pourquoi, pour améliorer le système, il faut ajouter des états. Comme le montre la figure 6.1, le problème vient du fait que le modèle ne sait pas, si une partie du lot de kanbans en production, a déjà été produite. Ce qui fait que les états $(X = 0, Y)$ sont certainement trop souvent visités. En effet, pour peu que Y soit suffisamment élevé, le taux λ est plus grand que μ/Y . Un kanban est donc beaucoup plus vite consommé que produit! Si tel est le cas, le taux de pénurie moyen est vraiment surestimé. Ainsi que l'explique la figure 6.2, on démultiplie les états (X, Y) tels que

$Y \geq \min(NE, 2)$. Pour ce faire on introduit un troisième paramètre Z qui est égal au nombre de kanbans déjà produits dans le lot. On a la relation suivante: $Z \leq (Y - 1)$; en effet, l'état (X, Y, Y) et l'état $(X + Y, 0, 0)$ sont confondus. Dans ce nouveau modèle, les transitions sont fortement simplifiées. Si l'on se trouve dans un état tel que $Y \neq 0$, alors il n'existe que deux solutions pour en sortir: consommer ou produire un *unique* kanban.

Le modèle témoigne encore d'une autre faiblesse: il ne sait pas accumuler le travail en retard. C'est-à-dire que, lorsqu'il y a pénurie, les commandes existantes sont perdues. Ce n'est, pour l'instant, pas un problème, car la demande est un processus continu, mais, si l'on veut proposer une modélisation concurrentielle (par rapport à une simulation, par exemple), il faut en tenir compte. Une autre amélioration qui apparaît évidente, serait de généraliser ce modèle *OSTOS* aux modèles *MSTOS* et *MSTMS*. Enfin, on peut également considérer les temps de production et de consommation, non plus comme des variables exponentielles, mais comme des sommes de variables exponentielles, afin de faire tendre leur distribution vers la loi normale $N(0, 1)$.

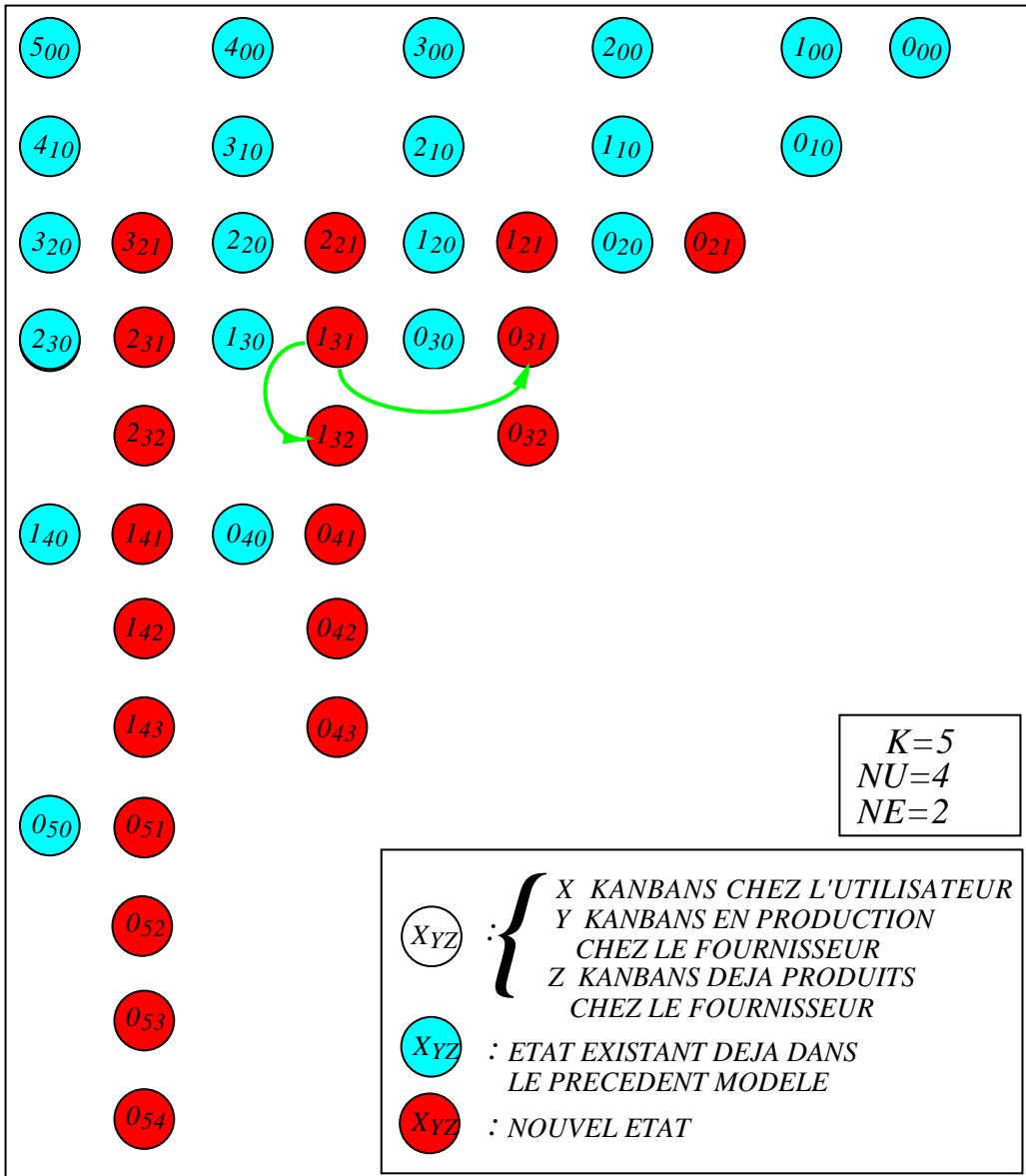


FIG. 6.2: Nouvel ensemble des états

Chapitre 7

Conclusion

Le développement d'un modèle de décision en gestion de production est un vaste projet. Au cours de ces quatre mois, j'ai découvert une application mathématique à l'industrie que je ne connaissais pas. Même si l'essence d'un système Kanban est relativement simple, les modèles que j'ai catalogués le sont nettement moins, et la littérature associée plus qu'obscur. Parfois, alors que je pensais avoir compris une heuristique, j'ai buté sur des graphiques, soit-disant explicatifs, en désaccord avec l'idée que je m'en était faite. Toutefois, cette diversité et cette complexité forment la richesse même des systèmes Kanbans. Elles reflètent bien les difficultés à affronter, lorsque l'on veut modéliser le fonctionnement du monde de l'industrie, aussi vaste qu'hétéroclite. Autant dire que le développement du nouveau modèle, même si sa base est le fruit de M. Wang et Wang, n'a pas été chose aisée. Toutefois, il m'a permis de me familiariser avec un domaine plus qu'intéressant ainsi qu'avec différents logiciels : *Matlab 5.3.0* pour la partie codage et calculs ; *Latex* et *Illustrator.5* pour la mise en page du document dont vous achevez la lecture. Ce qui a fait également la richesse et l'intérêt de ce projet est le fait de pouvoir créer quelque chose de neuf. Se demander, si le modèle que l'on a développé, va nous fournir des résultats sensés et soutenir la comparaison avec celui de Wang et Wang, a été réellement passionnant. Le fait qu'il reste encore tant de choses à découvrir et à modéliser me plaît particulièrement. Sans vouloir aucunement discréditer le travail que j'ai accompli (ce n'est pas à moi de le faire, s'il y a lieu!), il reste encore des pistes à exploiter. J'espère que ce projet aura une suite et que j'en ferai partie. Le prochain challenge étant de le confronter à la simulation élaborée par le LGPP-DGM. Je terminerai en remerciant le Professeur Th. M. Liebling et son assistant M. Jean-François Hêche de m'avoir proposé ce sujet, et tout particulièrement ce dernier pour l'aide et le soutien apporté.

Annexe A

Codes utilisés dans *Matlab 5.3.0*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Declaration des variables(ici, leurs valeurs sont
% les noms qui les representent le plus frequemment
% dans les fonctions suivantes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K=k;           % Nombre de kanbans total
NE=ne ou x;   % Niveau d'engagement
NU=nu ou y;   % Niveau d'urgence
Setup=set;    % Temps de setup moyen
Lambda=lambda; % Taux de consommation
Mu=mu;        % Taux de production/remplissage
max_penurie=tomax; % Pourcentage maximal autorise
               % de penurie
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function numero=trouver_numero(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction calcule le numero de la ligne(respectivement
% de la colonne)associee a l'etat determine par x et y.
% x et y representent le nombres de kanbans chez le consommateur
% et chez le fournisseur(respectivement).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if k-x-y==0
    numero=y+1;
```

```

else
    numero=(k-x-y)*(k+x+y+3)/2+y+1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function p=prob_de_production(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction definit la politique de
% production appliquee entre le niveau
% d'engagement et celui d'urgence(x et y).
% Elle peut, comme c'est le cas ici (no 4),
% prendre la valeur des fonctions suivantes:
% _ prob_de_production1.m
% _ prob_de_production2.m
% _ prob_de_production3.m
% _ prob_de_production4.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p=sparse(1,k);
for i=x:1:y-1
    p(1,i)=log(i)/log(y);
end;
for i=y:1:k
    p(1,i)=1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function p=prob_de_production1(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dans cette politique, la probabilite
% de production croit lineairement avec
% le nombre de kanbans.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p=sparse(1,k);
j=1/(y-x+1);
p(1,x)=j;

```

```
for i=x+1:1:y-1
    p(1,i)=p(1,i-1)+j;
end;
for i=y:1:k
    p(1,i)=1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function p=prob_de_production2(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dans cette politique, la probabilite
% de production croit exponentiellement
% avec le nombre de kanbans.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
p=sparse(1,k);
for i=x:1:y-1
    p(1,i)=exp((1/2)*i)/exp((1/2)*y);
end;
for i=y:1:k
    p(1,i)=1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function p=prob_de_production3(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dans cette politique, la probabilite
% de production croit de maniere
% logarithmique avec
% le nombre de kanbans.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
p=sparse(1,k);
for i=x:1:y-1
    p(1,i)=log(i)/log(y);
end;
for i=y:1:k
```

```
p(1,i)=1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function p=prob_de_production4(k,x,y)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dans cette politique, la probabilite
% de production est constante egale a
% 1/2, quelque soit le nombre de
% kanbans compris entre NE et NU-1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

p=sparse(1,k);
for i=x:1:y-1
    p(1,i)=.5;
end;
for i=y:1:k
    p(1,i)=1;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function matrix=trouver_la_matrice(k,ne,nu,lambda,mu,prob,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction fait tout d'abord appel a la fonction
% remplir_la_matrice.m. Ensuite, elle remplace les termes
% de la premiere colonne par 1. Elle inverse la matrice
% (pour pouvoir resoudre Ax=b et non pas xA=b)
% et la rend creuse.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

matrix=remplir_la_matrice(k,ne,nu,lambda,mu,prob,set);
for i=1:(((k+1)*(k+2)/2)-1)
    matrix(i,1)=1;
end;
matrix=matrix';
matrix=sparse(matrix);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Pi=trouver_la_distr(k,ne,nu,lambda,mu,prob,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction fait appel a la fonction trouver_la_matrice.m
% pour qu'elle lui fournisse la matrice de transition. Elle
% definit un second membre b comme une matrice creuse dont
% l'unique terme non-nul est le premier, egal a 1. Puis, elle
% resoud le systeme Ax=b et renvoie la distribution trouvee.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A=trouver_la_matrice(k,ne,nu,lambda,mu,prob,set);
b=sparse(((k+1)*(k+2)/2)-1,1);
b(1,1)=1;
Pi=A\b;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function B=matrice_etat_distr(k,ne,nu,lambda,mu,prob,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction fait appel a la fonction trouver_la_distr.m
% qui lui fournit la distribution stationnaire du systeme.
% Ensuite, a partir du numero de la ligne de la matrice de
% transition, elle determine le nombre de kanbans chez le
% consommateur et chez le fournisseur. Elle construit une
% matrice, dans laquelle est stocke:
% -premiere ligne: le nombre de kanbans chez le consommateur
% -deuxieme ligne: le nombre de kanbans chez le fournisseur
% -troisieme ligne: la probabilite d'etre dans cet etat.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=trouver_la_distr(k,ne,nu,lambda,mu,prob,set);
colonne=1;
decalage=0;
retrait=0;
for j=1:(((k+1)*(k+2)/2)-1)
    for i=1:colonne
        decalage=decalage+(k+1-i+1);
    end;

```

```
if j<=decalage
    for r=1:colonne-1
        retrait=retrait+k+1-r+1;
    end;
    B(2,j)=j-1-retrait;
    B(1,j)=k-B(2,j)-colonne+1;
    B(3,j)=x(j);
    decalage=0;
    retrait=0;
else
    colonne=colonne+1;
    for r=1:colonne-1
        retrait=retrait+k+1-r+1;
    end;
    B(2,j)=j-1-retrait;
    B(1,j)=k-B(2,j)-colonne+1;
    B(3,j)=x(j);
    decalage=0;
    retrait=0;
end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function e=stock_penurie_moyens(k,matrice)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction calcule le niveau de stock moyen
% ainsi que le taux de penurie moyen, associes a
% une distribution stationnaire.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

declaration_var;
for i=1:(((k+1)*(k+2)/2)-1)
    if matrice(1,i)|0
        vect1(1,i)=(matrice(1,i)-1/2)*matrice(3,i);
        vect2(1,i)=0;
    else
        vect1(1,i)=0;
        vect2(1,i)=matrice(3,i);
    end;
end;
```

```

end;
e(1,1)=sum(vect1);
e(1,2)=sum(vect2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function z0=remplir_graph1(k,lambda,mu,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction calcule pour  $1 \leq NE \leq K$  et pour
%  $1 \leq NU \leq K$ , le niveau moyen du stock associe.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for x0=1:k
    for y0=1:k
        if x0 <= y0
            if x0 == k
                z0(y0,x0) = ((k - (1/2)) * (k * mu / (lambda * (k + mu * set) + k * mu)));
            else
                prob = prob_de_production(k,x0,y0);
                m = matrice_etat_distr(k,x0,y0,lambda,mu,prob,set);
                e = stock_penurie_moyens(k,m);
                z0(y0,x0) = e(1,1);
            end;
        else;
        end;
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function graphique1(k,lambda,mu,tomax,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction effectue les graphiques
% du niveau de stock moyen et du taux de
% penurie, en fonction de NE et NU. Le
% nombre de kanbans K est fixe. Elle
% renvoie egalement les minima de ces
% deux graphiques et leurs arguments

```

```
% respectifs.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=(1:1:k)';
y=(1:1:k)';
[X,Y]=meshgrid(x,y);
z=rempkir_graph1(k,lambda,mu,set);
w=rempkir_graph1pen(k,lambda,mu,set);
subplot(1,2,1);
surf(x,y,z);
subplot(1,2,2);
surf(x,y,w);
grid;
a1=0;
o1=0;
for i=1:k
    for j=1:k
        if (z(i,j)==0)
            z(i,j)=k;
        else;
        end;
    end;
end;
m1=min(z);
m2=min(m1);
for l=1:k
    for n=1:k
        if z(l,n)==m2
            a1=1;
            o1=n;
        else;
        end;
    end;
end;
a2=0;
o2=0;
for i=1:k
    for j=1:k
        if (w(i,j)>tomax)
            z(i,j)=k;
        else;

```

```

        end;
    end;
end;
m3=min(z);
m4=min(m3);
for l=1:k
    for n=1:k
        if z(1,n)==m4
            a2=1;
            o2=n;
        else;
            end;
        end;
    end;
end;
m(1,1)=m2;
m(1,2)=o1;
m(1,3)=a1;
m(2,1)=m4;
m(2,2)=o2;
m(2,3)=a2;
stock_min_et_min_sans_penurie=m;
stock_min_et_min_sans_penurie

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function graphique2(k,lambda,mu,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction effectue les graphiques
% du niveau de stock moyen et du taux de
% penurie, en fonction de K. Les niveaux
% NE et NU sont fixes, arbitrairement, a
% la valeur entiere superieure du tiers
% et des deux-tiers de K. Remarque: cela
% peut changer.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=(1:1:k);
for k0=1:k
    if k0==1
        y(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));

```

```
        z(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
    else
        k1=k0/3;
        k2=2*k0/3;
        e=ceil(k1);
        u=ceil(k2);
        proba=prob_de_production(k0,e,u);
        m=matrice_etat_distr(k0,e,u,lambda,mu,proba,set);
        e=stock_penurie_moyens(k0,m);
        y(k0)=e(1,2);
        z(k0)=e(1,1);
    end;
end;
subplot(1,2,1);
plot(x,z);
subplot(1,2,2);
plot(x,y);
grid;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function graphique3(k,lambda,mu,tomax,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction effectue le meme travail
% que la fonction graphique1, mais avec
% toutes les politiques de production.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=(1:1:k)';
y=(1:1:k)';
[X,Y]=meshgrid(x,y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 1:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for x0=1:k
    for y0=1:k
        if x0<=y0
            if x0==k
```

```

        z1(y0,x0)=((k-(1/2))*(k*mu/(lambda*(k+mu*set)+k*mu)));
        w1(y0,x0)=(lambda*(k+mu*set)/(lambda*(k+mu*set)+k*mu));
    else
        prob1=prob_de_production1(k,x0,y0);
        m1=matrice_etat_distr(k,x0,y0,lambda,mu,prob1,set);
        e1=stock_penurie_moyens(k,m1);
        z1(y0,x0)=e1(1,1);
        w1(y0,x0)=e1(1,2);
    end;
else
    z1(y0,x0)=inf;
    w1(y0,x0)=inf;
end;
end;
end;

for i=1:k-1
    z1(i,i+1)=(z1(i,i)+z1(i+1,i+1))/2;
    w1(i,i+1)=(w1(i,i)+ w1(i+1,i+1))/2;
end;

subplot(4,2,1);
grid;
surf(x,y,z1);
xlabel('NE');
ylabel('NU');
title('Niveau de stock moyen');
subplot(4,2,2);
grid;
surf(x,y,w1);
xlabel('NE');
ylabel('NU');
title('Taux de penurie moyen');

for i=1:k-1
    z1(i,i+1)=inf;
    w1(i,i+1)=inf;
end;

l11=min(z1);
minimum11=min(l11);

```

```
for l=1:k
    for n=1:k
        if (z1(l,n)==minimum11)
            a11=1;
            o11=n;
        else;
            end;
        end;
    end;
end;
for i=1:k
    for j=1:k
        if (w1(i,j)>tomax)
            z1(i,j)=k;
        else;
            end;
        end;
    end;
end;
l12=min(z1);
minimum12=min(l12);
for l=1:k
    for n=1:k
        if z1(l,n)==minimum12
            a12=1;
            o12=n;
        else;
            end;
        end;
    end;
end;
lin(1,1)=minimum11;
lin(1,2)=o11;
lin(1,3)=a11;
lin(2,1)=minimum12;
lin(2,2)=o12;
lin(2,3)=a12;
politique_lineaire=lin;
politique_lineaire

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 2:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

for x0=1:k
  for y0=1:k
    if x0<=y0
      if x0==k
        z2(y0,x0)=((k-(1/2))*(k*mu/(lambda*(k+mu*set)+k*mu)));
        w2(y0,x0)=(lambda*(k+mu*set)/(lambda*(k+mu*set)+k*mu));
      else
        prob2=prob_de_production2(k,x0,y0);
        m2=matrice_etat_distr(k,x0,y0,lambda,mu,prob2,set);
        e2=stock_penurie_moyens(k,m2);
        z2(y0,x0)=e2(1,1);
        w2(y0,x0)=e2(1,2);
      end;
    else
      z2(y0,x0)=inf;
      w2(y0,x0)=inf;
    end;
  end;
end;

for i=1:k-1
  z2(i,i+1)=(z2(i,i)+z2(i+1,i+1))/2;
  w2(i,i+1)=(w2(i,i)+ w2(i+1,i+1))/2;
end;

subplot(4,2,3);
grid;
surf(x,y,z2);
xlabel('NE');
ylabel('NU');
subplot(4,2,4);
grid;
surf(x,y,w2);
xlabel('NE');
ylabel('NU');

for i=1:k-1
  z2(i,i+1)=inf;
  w2(i,i+1)=inf;
end;

```

```
l21=min(z2);
minimum21=min(l21);
for l=1:k
    for n=1:k
        if (z2(l,n)==minimum21)
            a21=1;
            o21=n;
        else;
            end;
        end;
    end;
end;
for i=1:k
    for j=1:k
        if (w2(i,j)>tomax)
            z2(i,j)=k;
        else;
            end;
        end;
    end;
end;
l22=min(z2);
minimum22=min(l22);
for l=1:k
    for n=1:k
        if z2(l,n)==minimum22
            a22=1;
            o22=n;
        else;
            end;
        end;
    end;
end;
expo(1,1)=minimum21;
expo(1,2)=o21;
expo(1,3)=a21;
expo(2,1)=minimum22;
expo(2,2)=o22;
expo(2,3)=a22;
politique_exponentielle=expo;
politique_exponentielle

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 3:
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for x0=1:k
    for y0=1:k
        if x0<=y0
            if x0==k
                z3(y0,x0)=((k-(1/2))*(k*mu/(lambda*(k+mu*set)+k*mu)));
                w3(y0,x0)=(lambda*(k+mu*set)/(lambda*(k+mu*set)+k*mu));
            else
                prob3=prob_de_production3(k,x0,y0);
                m3=matrice_etat_distr(k,x0,y0,lambda,mu,prob3,set);
                e3=stock_penurie_moyens(k,m3);
                z3(y0,x0)=e3(1,1);
                w3(y0,x0)=e3(1,2);
            end;
        else
            z3(y0,x0)=inf;
            w3(y0,x0)=inf;
        end;
    end;
end;

for i=1:k-1
    z3(i,i+1)=(z3(i,i)+z3(i+1,i+1))/2;
    w3(i,i+1)=(w3(i,i)+ w3(i+1,i+1))/2;
end;

subplot(4,2,5);
grid;
surf(x,y,z3);
xlabel('NE');
ylabel('NU');
subplot(4,2,6);
grid;
surf(x,y,w3);
xlabel('NE');
ylabel('NU');

for i=1:k-1
    z3(i,i+1)=inf;
    w3(i,i+1)=inf;
```

```
end;

l31=min(z3);
minimum31=min(l31);
for l=1:k
    for n=1:k
        if (z3(l,n)==minimum31)
            a31=l;
            o31=n;
        else;
            end;
        end;
    end;
end;
for i=1:k
    for j=1:k
        if (w3(i,j)>tomax)
            z3(i,j)=k;
        else;
            end;
        end;
    end;
end;
l32=min(z3);
minimum32=min(l32);
for l=1:k
    for n=1:k
        if z3(l,n)==minimum32
            a32=l;
            o32=n;
        else;
            end;
        end;
    end;
end;
loga(1,1)=minimum31;
loga(1,2)=o31;
loga(1,3)=a31;
loga(2,1)=minimum32;
loga(2,2)=o32;
loga(2,3)=a32;
politique_logarithmique=loga;
politique_logarithmique
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 4:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for x0=1:k
    for y0=1:k
        if x0<=y0
            if x0==k
                z4(y0,x0)=((k-(1/2))*(k*mu/(lambda*(k+mu*set)+k*mu)));
                w4(y0,x0)=(lambda*(k+mu*set)/(lambda*(k+mu*set)+k*mu));
            else
                prob4=prob_de_production4(k,x0,y0);
                m4=matrice_etat_distr(k,x0,y0,lambda,mu,prob4,set);
                e4=stock_penurie_moyens(k,m4);
                z4(y0,x0)=e4(1,1);
                w4(y0,x0)=e4(1,2);
            end;
        else
            z4(y0,x0)=inf;
            w4(y0,x0)=inf;
        end;
    end;
end;

for i=1:k-1
    z4(i,i+1)=(z4(i,i)+z4(i+1,i+1))/2;
    w4(i,i+1)=(w4(i,i)+ w4(i+1,i+1))/2;
end;

subplot(4,2,7);
grid;
surf(x,y,z4);
xlabel('NE');
ylabel('NU');
subplot(4,2,8);
grid;
surf(x,y,w4);
xlabel('NE');
ylabel('NU');

```

```
for i=1:k-1
    z4(i,i+1)=inf;
    w4(i,i+1)=inf;
end;

l41=min(z4);
minimum41=min(l41);
for l=1:k
    for n=1:k
        if (z4(l,n)==minimum41)
            a41=l;
            o41=n;
        else;
            end;
        end;
    end;
end;
for i=1:k
    for j=1:k
        if (w4(i,j)>tomax)
            z4(i,j)=k;
        else;
            end;
        end;
    end;
end;
l42=min(z4);
minimum42=min(l42);
for l=1:k
    for n=1:k
        if z4(l,n)==minimum42
            a42=l;
            o42=n;
        else;
            end;
        end;
    end;
end;
cons(1,1)=minimum41;
cons(1,2)=o41;
cons(1,3)=a41;
cons(2,1)=minimum42;
cons(2,2)=o42;
cons(2,3)=a42;
```

```

politique_constante=cons;
politique_constante

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function graphique4(k,lambda,mu,set)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Cette fonction effectue le meme travail
% que la fonction graphique2, mais avec
% toutes les politiques de production et
% le modele de Wang.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k0=1:k
    if k0==1
        y1(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));
        z1(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
    else
        k1=k0/3;
        k2=2*k0/3;
        e1=ceil(k1);
        u1=ceil(k2);
        proba1=prob_de_production1(k0,e1 ,u1);
        m1=matrice_etat_distr(k0,e1,u1,lambda,mu,proba1,set);
        ex1=stock_penurie_moyens(k0,m1);
        y1(k0)=ex1(1,2);
        z1(k0)=ex1(1,1);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k0=1:k
    if k0==1

```

```
        y2(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));
        z2(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
    else
        k1=k0/3;
        k2=2*k0/3;
        e2=ceil(k1);
        u2=ceil(k2);
        proba2=prob_de_production2(k0,e2,u2);
        m2=matrice_etat_distr(k0,e2,u2,lambda,mu,proba2,set);
        ex2=stock_penurie_moyens(k0,m2);
        y2(k0)=ex2(1,2);
        z2(k0)=ex2(1,1);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k0=1:k
    if k0==1
        y3(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));
        z3(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
    else
        k1=k0/3;
        k2=2*k0/3;
        e3=ceil(k1);
        u3=ceil(k2);
        proba3=prob_de_production3(k0,e3,u3);
        m3=matrice_etat_distr(k0,e3,u3,lambda,mu,proba3,set);
        ex3=stock_penurie_moyens(k0,m3);
        y3(k0)=ex3(1,2);
        z3(k0)=ex3(1,1);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Politique de production numero 4
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for k0=1:k
```

```

if k0==1
    y4(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));
    z4(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
else
    k1=k0/3;
    k2=2*k0/3;
    e4=ceil(k1);
    u4=ceil(k2);
    proba4=prob_de_production4(k0,e4,u4);
    m4=matrice_etat_distr(k0,e4,u4,lambda,mu,proba4,set);
    ex4=stock_penurie_moyens(k0,m4);
    y4(k0)=ex4(1,2);
    z4(k0)=ex4(1,1);
end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Modele de Wang et Wang
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k0=1:k
    if k0==1
        y5(k0)=(lambda*(k0+mu*set)/(lambda*(k0+mu*set)+k0*mu));
        z5(k0)=((k0-(1/2))*(k0*mu/(lambda*(k0+mu*set)+k0*mu)));
    else
        e5=1;
        u5=1;
        proba5=prob_de_production(k0,e5,u5);
        m5=matrice_etat_distr(k0,e5,u5,lambda,mu,proba4,set);
        ex5=stock_penurie_moyens(k0,m5);
        y5(k0)=ex5(1,2);
        z5(k0)=ex5(1,1);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Graphiques
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

x=(1:1:k);
subplot(1,2,1);
plot(x,z1,'b',x,z2,'r',x,z3,'g',x,z4,'c',x,z5,'k--');
grid;

```

```
xlabel('Nombre de kanbans K');
ylabel('Niveau de stock moyen');
legend('politique lineaire','politique exponentielle',
'politique logarithmique','politique constante','Wang et Wang',2);
subplot(1,2,2);
plot(x,y1,'b',x,y2,'r',x,y3,'g',x,y4,'c',x,y5,'k--');
grid;
xlabel('Nombre de kanbans K');
ylabel('Taux de penurie moyen');
legend('politique lineaire','politique exponentielle',
'politique logarithmique','politique constante','Wang et Wang',1);
```


Bibliographie

- [1] C.-C. HUANG and A.KUSIAK, *Overview of Kanban systems*, Int. J. Computer Integrated Manufacturing - Vol 9, 1996.
- [2] W. PRICE, M GRAVEL and A. L.NSAKANDA, *A review of optimisation models of Kanban-based production systems*, E. J. of Operational Research - No 75, 1994.
- [3] H. WANG and H. P. WANG *Optimum number of kanbans between two adjacent workstations in a JIT system*, Int. J. of Production Economics - No 22, 1991.
- [4] H. WANG and H. P. WANG, *Determining the number of kanbans: a step toward non-stock-production*, Int. J. of Production Research - Vol 28, 1990.
- [5] J.-F. HÊCHE et Th. M. LIEBLING, *Processus de Markov à temps continu*, ROSO-DMA-EPFL, 1999.
- [6] P. CERF , *Développement d'un outil de simulation pour un système Kanban*, Projet de semestre LGPP-DGM-EPFL, 1999.
- [7] M.S. GOCKENBACH, *A Practical introduction to Matlab 5.3.0*.