



Titre: Etude par simulation des performances de différentes règles d'ordonnement de la production sur commande avec machine à traitement par lot
Title:

Auteur: François Racine
Author:

Date: 1989

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Racine, F. (1989). Etude par simulation des performances de différentes règles d'ordonnement de la production sur commande avec machine à traitement par lot [Master's thesis, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/58275/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/58275/>
PolyPublie URL:

Directeurs de recherche:
Advisors:

Programme: Unspecified
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE PAR SIMULATION DES PERFORMANCES
DE DIFFÉRENTES RÈGLES D'ORDONNANCEMENT
DE LA PRODUCTION SUR COMMANDE AVEC
MACHINE À TRAITEMENT PAR LOT

par

François RACINE
DÉPARTEMENT DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU GRADE DE MAÎTRE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)

avril 1989

c François Racine 1989

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-52709-9

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Ce mémoire intitulé:

Étude par simulation des performances de différentes
règles d'ordonnancement de la production sur commande
avec machine à traitement par lot

Présenté par: François Racine

en vue de l'obtention du grade: M.Sc.A.

a été dûment accepté par le jury d'examen constitué de:

M. Laurent Villeneuve

M. Ing., président

M. Daniel Leblanc

Ph.D.

M. Chelliah Sriskandarajah

D.Sc.

SOMMAIRE

Le but de ce travail est d'étudier un cas spécifique de l'ordonnancement de production sur commande. On considère le cas où une machine, que nous appelons machine à traitement par lot, peut traiter plusieurs pièces en même temps. Cette machine à traitement par lot se caractérise aussi par des temps de traitement beaucoup plus longs que ceux des machines ordinaires.

Contrairement aux machines ordinaires, plusieurs types de décision doivent se prendre lorsque la machine à traitement par lot se libère. On doit choisir une famille de pièces à traiter et parmi cette famille quelles pièces formeront le lot. Nous avons aussi considéré le cas où une attente serait préférable avant de faire démarrer la machine à traitement par lot en vue de maximiser la productivité de celle-ci.

Nous avons créé un atelier de sous-traitance fictif comprenant 9 machines ordinaires et une machine à traitement par lot. Le langage de simulation SLAM II a été utilisé en vue de déterminer les performances à long terme de diverses règles de priorité.

Il en découle des expériences réalisées que, pour optimiser les performances globales de l'atelier, une combinaison de règles de priorité visant à utiliser au maximum la capacité de la machine à traitement par lot et qui considère les dates d'exigibilité des pièces est nécessaire. Quant à l'attente, elle tend à diminuer les

coûts d'utilisation de la machine à traitement par lot, cependant elle a un impact négatif sur les performances globales de l'atelier.

Mots-clés: ordonnancement, simulation, règles de priorité, familles, lots.

ABSTRACT

The purpose of this project is to study a specific case of the job shop scheduling problem. We consider the situation where a machine can execute many products at the same time. A characteristic of this type of machine is that its processing times may be much longer than those of the ordinary machines.

As opposed to the ordinary machines, different types of decisions must be taken when the machine that executes many products at the same time becomes free. We have to choose a family of products and within that family which products will form a batch. We also considered the case where it would be preferable to wait before starting the machine that can execute many products in order to maximise the productivity of this machine.

We created a small job shop including 9 ordinary machines and one that can execute many products at the same time. We used the simulation language SLAM II to determine the long-term performances of diverse dispatching rules.

From the experiments, we found that to optimise the global performances of the shop, we need a combination of dispatching rules that will try to maximise the capacity of the machine and that considers the due dates of the products. The waiting is beneficial if we want to lower the costs of using the machine but it has a negative effect on the global performances of the shop.

REMERCIEMENTS

Je désire exprimer ma reconnaissance à tous ceux qui ont collaboré de près ou de loin à la réalisation de ce mémoire.

Je tiens à remercier plus particulièrement mes directeurs de recherche, M. Daniel Leblanc, Ph.D, professeur agrégé au département de génie industriel et M. François Soumis, Ph.D., professeur titulaire au département de mathématiques appliquées. Je les remercie de m'avoir proposé le sujet de cette recherche, de m'avoir guidé lors de son élaboration et d'y avoir consacré de nombreuses heures.

Je remercie aussi M. Laurent Villeneuve, M.Ing., directeur du département de génie industriel et M. Chelliah Sriskandarajah, D.Sc., associé de recherche au département de mathématiques appliquées d'avoir agi respectivement comme président du jury et membre du jury. Je les remercie pour les commentaires qu'ils ont apportés à mon travail.

Je remercie aussi cordialement M. Heang Ngai Cheng, associé de recherche au département de génie industriel, pour tous les conseils qu'il m'a donnés lors de la partie de programmation.

Je désire aussi exprimer ma gratitude au Conseil de Recherche en Sciences Naturelles et Génie du Canada de m'avoir octroyé une bourse d'études supérieures.

Je voudrais aussi remercier le bureau Traitement de Textes Marcil et Giguère enr. de s'être chargé de la dactylographie de ce mémoire.

Finalement, je voudrais remercier mes parents de m'avoir encouragé à étudier et de m'avoir offert un environnement propice à cela.

TABLE DES MATIÈRES

	page
SOMMAIRE	iv
ABSTRACT	vi
REMERCIEMENTS	vii
LISTE DES FIGURES	xi
LISTE DES TABLEAUX	xii
INTRODUCTION	1
CHAPITRE 1 L'ORDONNANCEMENT DE PRODUCTION DANS UN ATELIER DE SOUS-TRAITANCE	2
1.1 INTRODUCTION	2
1.2 DÉFINITIONS	3
1.3 TERMINOLOGIE	5
1.4 MESURES DE PERFORMANCE	7
CHAPITRE 2 REVUE DE LITTÉRATURE	10
2.1 INTRODUCTION	10
2.2 TYPES DE PROBLÈMES	10
2.3 L'ATELIER DE SOUS-TRAITANCE GÉNÉRAL	14
2.4 HYPOTHÈSES DE BASE ET RESTRICTIONS	15
2.5 RÈGLES DE PRIORITÉ	17
2.6 RÉSULTATS DES RECHERCHES	19
2.7 RELAXATION DE CERTAINES HYPOTHÈSES ET RESTRICTIONS	23
2.8 CONCLUSION	24
CHAPITRE 3 LA PROBLÉMATIQUE DE LA MACHINE À TRAITEMENT PAR LOT	26
3.1 INTRODUCTION	26
3.2 TYPES DE PROCÉDÉS	26
3.3 INTÉRÊT DE L'ÉTUDE	27
3.4 ATELIER TYPE	28
3.5 HYPOTHÈSES ET RESTRICTIONS DE BASE	30
3.6 ORDONNANCEMENT DES MACHINES ORDINAIRES	31
3.7 FACTEURS DISCRIMINANTS POUR LA MACHINE À TRAITEMENT PAR LOT	32
3.8 CLASSIFICATION DES FACTEURS	34
3.9 DÉCISIONS À PRENDRE LORS DE L'ORDONNANCEMENT	36
3.10 OBJET DE L'ÉTUDE	37

CHAPITRE 4	ORDONNANCEMENT DE LA MACHINE À TRAITEMENT PAR LOT	38
4.1	INTRODUCTION	38
4.2	SCHÉMATISATION DE LA MACHINE À TRAITEMENT PAR LOT	38
4.3	RÈGLES DE PRIORITÉ ÉTUDIÉES	40
4.4	PRISE DE DÉCISIONS	46
CHAPITRE 5	PLAN D'EXPÉRIENCE	48
5.1	INTRODUCTION	48
5.2	CHOIX DE LA FAMILLE ET DES PIÈCES SEULEMENT	49
5.3	POSITION DE LA MACHINE À TRAITEMENT PAR LOT	51
5.4	POLITIQUE D'ATTENTE POUR LA MACHINE À TRAITEMENT PAR LOT	52
5.5	PARAMÈTRES DES PIÈCES À EXÉCUTER	52
5.6	STATISTIQUES	54
CHAPITRE 6	PROGRAMMATION	56
6.1	INTRODUCTION	56
6.2	CHOIX DU LANGAGE	56
6.3	ARRIVÉE DES COMMANDES	59
6.4	ASSIGNATION DES ATTRIBUTS	59
6.5	SOUS-PROGRAMMES	62
6.6	FORMATION DES LOTS	64
6.7	DURÉE DE SIMULATION	64
CHAPITRE 7	ANALYSE DES RÉSULTATS	66
7.1	INTRODUCTION	66
7.2	ANALYSE DES COMBINAISONS DE RÈGLES	66
7.2.1	RANDOM-RANDOM	67
7.2.2	DUEDATE-SPT	69
7.2.3	DUEDATE-AVGDUEDATE, DUEDATE-WORKLOAD ET SLACK-WORKLOAD	70
7.3	POSITION EN AVAL DE LA MACHINE À TRAITEMENT PAR LOT	78
7.4	POLITIQUE D'ATTENTE	79
7.5	RECOMMANDATIONS	85
CONCLUSION	87
BIBLIOGRAPHIE	89
ANNEXE	92

LISTE DES FIGURES

Fig. 1.1	Atelier de sous-traitance	4
Fig. 2.1	Situations possibles	11
Fig. 3.1	Atelier de sous-traitance	29
Fig. 4.1	Atelier de sous-traitance général avec machines ordinaires seulement	39
Fig. 4.2	Machine à traitement par lot	39
Fig. 4.3	Productivité en fonction du temps d'attente	44
Fig. 6.1	Logique de programmation	58
Fig. 7.1	Temps en atelier avec RANDOM-RANDOM	68
Fig. 7.2	Temps en atelier	71
Fig. 7.3	Écart exigibilité-terminaison	73
Fig. 7.4	Pourcentage de pièces en retard	75
Fig. 7.5	Nombre de lots traités	76

LISTE DES TABLEAUX

	page
Tableau 2.1 Résultats de certaines règles en fonction du nombre moyen de commandes dans l'atelier	20
Tableau 2.2 Résultats de TSPT et RSPT en fonction du nombre moyen de commandes dans atelier	21
Tableau 2.3 Résultats de quelques règles en fonction de critères ayant rapport aux dates d'exigibilité	22
Tableau 7.1 Performances par familles de pièces selon DUEDATE-WORKLOAD	78
Tableau 7.2 Comparaison des performances avec machine à traitement par lot en 8 ^{ième} position	80
Tableau 7.3 Taux de remplissage en fonction du temps de traitement moyen	81
Tableau 7.4 Comparaison des résultats de la combinaison DUEDATE-WORKLOAD en utilisant la formule Prod 1	82
Tableau 7.5 Comparaison des résultats des combinaisons DUEDATE-WORKLOAD et SLACK-WORKLOAD en considérant l'attente selon Prod 2	84

INTRODUCTION

Dans les ateliers de sous-traitance survient fréquemment le problème de construire des horaires de production les plus efficaces possible. On désire des horaires qui respectent le plus possible les dates de livraison exigées par les clients, qui minimisent les stocks de produits en cours, etc.

La majorité des recherches faites sur l'ordonnancement de production considère l'atelier comme un réseau de files d'attente (une par machine). Lorsqu'une machine se libère, une décision se prend à savoir quel produit exécuter par la suite. Cependant, ces recherches considèrent qu'une machine ne peut exécuter qu'une seule pièce à la fois. On éliminera cette contrainte pour étudier le cas où une machine peut traiter plusieurs pièces différentes ou non à la fois. L'intérêt d'étudier cette machine vient du fait que son temps d'exécution est généralement long, elle peut représentée un goulot d'étranglement, son coût d'utilisation peut être élevé, etc.

On cherchera donc à trouver une méthode d'ordonnancement pour la machine à traitement par lot en particulier. Pour ce faire on utilisera le même outil qui a servi lors des études antérieures soit la simulation en vue de déterminer les performances à long terme de l'atelier.

CHAPITRE 1

L'ORDONNANCEMENT DE PRODUCTION DANS UN ATELIER DE SOUS-TRAITANCE

1.1 INTRODUCTION

Dans les environnements manufacturiers, plusieurs types de décision doivent se prendre: location d'usines et de centres de distribution, contrôle d'inventaires, planification, ordonnancement de production, etc. Certaines de ces décisions se font à un haut niveau dans la hiérarchie et ont une perspective à long-terme. L'ordonnancement de production se situe au bas de cette hiérarchie et a pour but d'assigner les différentes opérations sur les diverses machines et ce pour un intervalle de temps assez court et en tenant compte de la capacité de production disponible. De plus pour ce faire, on a besoin d'information détaillée. La fonction d'ordonnancement joue un rôle important car de nombreux bénéfices peuvent en résulter; coûts plus faibles, meilleur respect des demandes des clients, etc. Le problème d'ordonnancement de production dans un atelier de sous-traitance est mieux connu sous son nom anglais: job shop scheduling. Il s'agit d'un problème classique qui a été l'objet de nombreuses recherches.

1.2 DÉFINITIONS

Avant d'élaborer sur l'ordonnancement de production dans un atelier de sous-traitance, il semble nécessaire de bien définir certains termes et paramètres qui seront utilisés subséquemment. Premièrement, un atelier de sous-traitance exécute des commandes provenant de donneurs d'ordres selon les directives ou en accord avec ces derniers [1]. Une commande consistant à réaliser un produit en quantité voulue. Les donneurs d'ordres sont habituellement d'autres entreprises n'ayant pas la compétence nécessaire pour réaliser les travaux: c'est ce qu'on appelle la sous-traitance de spécialité. Ou bien ils dépassent leur capacité de production et doivent faire réaliser des travaux à l'extérieur: c'est la sous-traitance de capacité. Peu importe le type de sous-traitance, l'atelier se doit d'être en mesure de réaliser une large gamme de produits. Pour assurer cette versatilité, l'atelier comportera plusieurs machines généralement différentes soit par leur type soit par leur mode de fonctionnement: les temps d'exécution, de mise en route, le nombre de produits pouvant être réalisés en même temps, etc peuvent différer considérablement d'une machine à l'autre. Pour réaliser un produit on aura besoin d'une ou de plusieurs opérations sur une ou plusieurs de ces machines. Les cheminements dans l'atelier varient considérablement d'un produit à l'autre soit par le nombre ou par l'ordre par lequel ces machines seront utilisées. De plus certains produits peuvent avoir plusieurs cheminements possibles.

Un atelier de sous-traitance avec quelques cheminements possibles est représenté par la figure suivante:

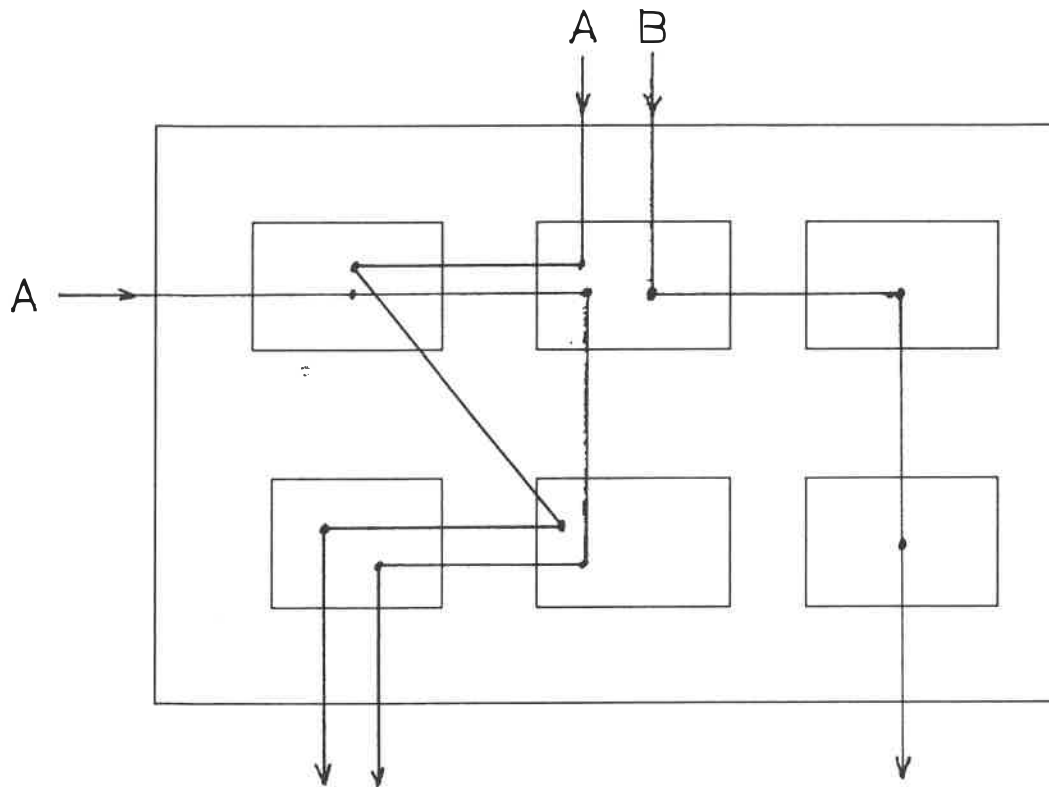


Fig. 1.1

Atelier de sous-traitance

Cet atelier comprend 6 machines et on y a représenté le cheminement de 2 produits, On note aussi qu'il peut y avoir plusieurs cheminements possibles pour un produit (ex.: A).

Deuxièmement, l'ordonnancement consiste à déterminer un horaire de production pour chacune des commandes sur les différentes machines de l'atelier. Cet horaire doit respecter le cheminement de chacun des produits, la durée de chaque opération, la capacité des machines en nombre de produits traités simultanément et possiblement d'autres contraintes imposées par l'atelier. On

désirera un horaire respectant le mieux possible les dates d'exigibilité, diminuant les niveaux de produits en cours, les temps d'attente des produits ou les temps morts des machines, etc.

1.3 TERMINOLOGIE

Définissons maintenant certains paramètres utilisés en ordonnancement de production.

i commande (job): séquence d'opérations inter-reliées par des contraintes de précédence en vue de réaliser un produit ($i = 1, 2 \dots n$).

machine: pièce d'équipement capable d'effectuer un type d'opération (m machines).

j opération: tâche élémentaire effectuée sur une machine.

P_{ij} temps de traitement (processing time): durée de l'opération j de la commande i .

R_i temps d'arrivée (arrival time): temps à partir duquel la première opération d'une commande i peut débiter.

D_i date d'exigibilité (due date): date à laquelle l'atelier doit avoir complété la commande i .

C_i temps de terminaison (completion time): temps auquel une commande i est terminée.

W_{ij} temps d'attente (waiting time): temps d'attente de l'opération j de la commande i .

F_i temps en atelier (flow time): temps qu'une commande i passe dans l'atelier:
 $F_i = C_i - R_i$.

L_i écart exigibilité-terminaison (lateness): différence entre le temps de terminaison et la date d'exigibilité d'une commande i :

$$L_i = C_i - D_i$$

T_i retard (tardiness):

$$T_i = \max \{L_i, 0\}.$$

E_i avance (earliness):

$$E_i = \max \{-L_i, 0\}.$$

1.4 MESURES DE PERFORMANCE

La tâche du responsable de l'ordonnancement de la production consiste à utiliser les paramètres qui lui sont connus et de construire un horaire de production.

Suite à la construction d'un horaire on connaîtra les paramètres suivants:

- temps de terminaison;
- temps d'attente;
- temps en atelier;
- écarts exigibilité-terminaison;
- retards;
- avances.

Étant donné qu'on peut construire plusieurs horaires de production avec un même ensemble de commandes, on a besoin de certains critères d'évaluation pour les comparer, c'est ce qu'on appelle les mesures de performance, elles sont de trois types: voir [2] [3] et [4].

- Mesures basées sur le temps de terminaison:

. temps moyen en atelier $\bar{F} = \frac{1}{n} \sum_i F_i$

. temps maximum en atelier $F_{\max} = \max_i \{F_i\}$

. temps moyen de terminaison $\bar{C} = \frac{1}{n} \sum_i C_i$

. temps maximum de terminaison $C_{\max} = \max_i \{C_i\}$

- Mesures basées sur les dates d'exigibilité:

$$\text{. retard moyen } \bar{T} = \frac{1}{n} \sum_i T_i$$

$$\text{. retard maximum } T_{\max} = \max_i \{T_i\}$$

$$\text{. écart exigibilité-terminaison moyen } L = \frac{1}{n} \sum_i L_i$$

. nombre de commandes en retard:

$$N_r = \sum_i \delta(T_i) \quad \text{ou } \delta(T_i) = \begin{cases} 1 & \text{si } T_i > 0 \\ 0 & \text{sinon} \end{cases}$$

- Mesures basées sur les inventaires de produits en cours et sur l'utilisation des machines:

. utilisation moyenne des machines:

$$\bar{U} = \frac{\sum_i \sum_j P_{ij}}{n F_{\max}}$$

. nombre moyen de commandes dans l'atelier pour un intervalle de temps:

$$\bar{N}(t_i, t_j) = \frac{1}{t_j - t_i} \int_{t_i}^{t_j} N(t) dt$$

Notons qu'il existe d'autres mesures de performance et que seulement quelques unes sont utilisées pour comparer des horaires de production car certaines sont plus simples que d'autres à calculer et qu'il existe des relations entre les résultats de certaines. Le choix d'utiliser une mesure par rapport à une autre est spécifique à chaque atelier et dépend des objectifs visés.

Dans la section suivante, on présente les principaux résultats de la théorie de l'ordonnancement de production dans un atelier de sous-traitance qui ont été obtenus jusqu'à présent.

CHAPITRE 2

REVUE DE LITTÉRATURE

2.1 INTRODUCTION

En consultant la littérature sur la théorie de l'ordonnancement, on constate que peu de nouveautés sont apparues depuis une vingtaine d'années. En réalité, quoique la formulation du problème d'ordonnancement soit assez simple, il est extrêmement difficile à résoudre. Des solutions optimales sont possibles seulement pour des cas dégénérés et spécifiques ne représentant pas vraiment la réalité. Voyons maintenant les recherches qui ont été réalisées.

2.2 TYPES DE PROBLÈMES

Deux types de problèmes ont été étudiés lors de ces recherches; les problèmes statiques et dynamiques. Le cas statique consiste à produire un horaire permettant de compléter un ensemble de commandes disponibles à un certain temps t . Aucune autre commande ne s'ajoute à partir du temps t jusqu'à la fin de la période d'ordonnancement. Le cas dynamique considère que les commandes

arrivent continuellement et que l'ordonnancement est effectué en cours de production. Dans ce cas, les recherches ont analysé les performances à long terme de l'atelier. De plus ces recherches ont aussi considéré les cas où les données étaient déterministes ou stochastiques. Dans le cas déterministe, les temps d'arrivée des commandes et les temps d'exécution sur les machines sont tous connus. Dans le cas stochastique, les temps d'arrivée et/ou les temps d'exécution suivent une distribution probabiliste. La majorité des recherches considère que les temps inter-arrivée ainsi que les temps d'exécution suivent une distribution exponentielle.

Le graphique suivant représente l'ensemble des situations possibles ainsi que leurs méthodes de résolution respectives.

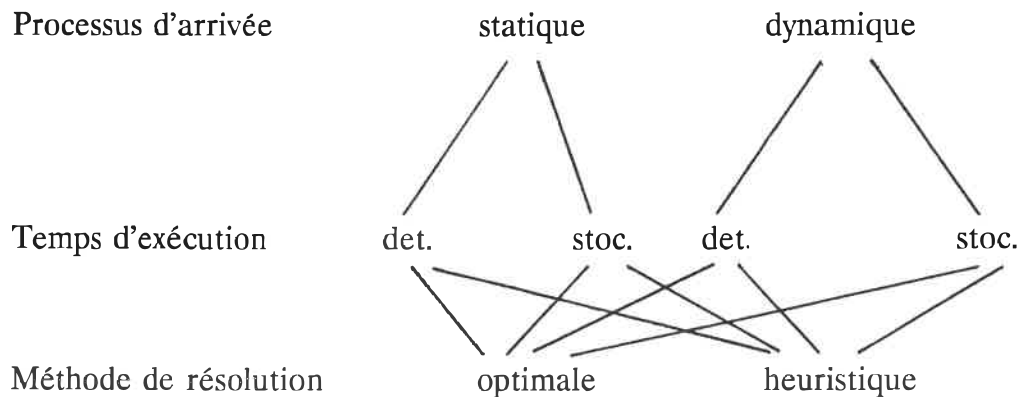


Fig 2.1

Situations possibles

Peu importe le cas, il est possible d'obtenir des solutions optimales ou heuristiques. Cependant l'optimalité est seulement possible pour des cas dégénérés ou spécifiques et cet optimum est obtenu pour quelques mesures de performance seulement, voir [4] tableau 5.11.

Regardons maintenant chacune des 4 situations possibles; statique déterministe, statique stochastique, dynamique déterministe et dynamique stochastique. Le cas statique déterministe est définitivement le plus simple. Il existe des algorithmes optimaux pour certaines mesures de performance et ce pour certains problèmes seulement, voir [5], [6], [7], [8], [9] et [10]. Quoique certains de ces problèmes sont très simples et ne sont que peut fréquemment rencontrés, les résultats obtenus peuvent servir de base de départ pour des problèmes plus complexes d'ordonnement.

Blazewicz [5] présente les résultats les plus importants concernant les problèmes d'ordonnement avec des machines en parallèle. Il y présente des méthodes d'optimisation ainsi que des algorithmes d'approximation ainsi qu'une évaluation de leurs performances. Il étudie le cas où plusieurs tâches doivent être exécutées sur plusieurs ordinateurs semblables.

Lawler [7] présente les principaux modèles déterministiques de l'ordonnement en mettant l'emphase sur ceux qui sont efficaces et qui sont bornés de façon polynomiale. Des algorithmes d'optimisation sont présentés au lieu

de méthodes approximatives. Il y présente des résultats pour les machines uniques, parallèles jusqu'à l'atelier général de sous-traitance.

Lawler, Lenstra et Rinnooy Kan [8] présentent aussi les principaux résultats des problèmes déterministes d'ordonnancement. Ils portent cependant l'attention sur les plus récents développements en ce qui a trait aux algorithmes de résolution et aux méthodes approximatives et les analysent en terme de degré de complexité (polynomiale, exponentielle).

Lenstra et Rinnooy Kan [9] présentent pour chaque type de problèmes, plusieurs publications apparues à partir de 1981. Ils ont classifié les problèmes selon plusieurs caractéristiques: nombre et type de machines, type d'atelier, caractéristiques spécifiques aux pièces, critères d'optimisation, etc. On y retrouve plusieurs algorithmes ou méthodes approximatives pour chaque catégorie.

En réalité peu de problèmes sont faciles à résoudre de façon optimale vue la complexité du problème. Carlier et Pinson [11] ont été les premiers à résoudre de façon optimale le cas avec 10 machines et 10 commandes et avec comme objectif principal de minimiser le temps total en atelier. En utilisant une méthode de "branch and bound" sur un mini-ordinateur, l'optimalité a nécessité 5 heures de CPU en 1988. Il semble donc préférable d'utiliser des méthodes heuristiques pour résoudre le problème d'ordonnancement. Elles sont surtout basées sur les règles de priorité (dispatching rules) permettant de sélectionner l'opération à

assigner à une machine qui devient libre.

Le cas statique stochastique peut être résolu par programmation dynamique stochastique mais cela n'est encore possible que pour de très petits problèmes. On fait encore souvent appel aux méthodes heuristiques pour résoudre le problème. Pinedo et Schrage [12] présente des résultats récents pour le domaine stochastique et ce pour plusieurs types d'atelier et d'objectifs. Ils y présentent aussi les différences et similitudes entre les divers modèles déterministes et stochastiques.

Les 2 derniers cas, dynamique déterministe et dynamique stochastique sont les plus difficiles à résoudre car nous retrouvons en régime permanent. Par contre, certaines règles de priorité peuvent optimiser certaines mesures de performance mais encore seulement pour des cas dégénérés. Dans le cas général on fait appel aux méthodes heuristiques.

2.3 L'ATELIER DE SOUS-TRAITANCE GÉNÉRAL

Notre problème porte sur l'atelier de sous-traitance général; c'est-à-dire que les commandes arrivent aléatoirement dans le temps, qu'il y a continuellement du travail dans l'atelier et que l'on a plusieurs machines. Une solution optimale est donc impossible. Les recherches ont considéré l'atelier comme un réseau de queues (devant les machines) et où les commandes circulent entre ces queues jusqu'à ce

qu'elles soient terminées. L'ordonnancement se fait donc à l'aide de règles de priorité (une par machine).

On présente maintenant les principaux résultats obtenus jusqu'à présent pour l'atelier de sous-traitance général. Ils ont été obtenus par simulation et l'on y a comparé plusieurs règles de priorité en fonction de diverses mesures de performance. Ces études ont supposé que les temps inter-arrivée et d'exécution suivaient des distributions exponentielles. Cependant les temps d'exécution sont connus lorsqu'une commande arrive à l'atelier. Nous sommes donc dans le cas dynamique stochastique.

2.4 HYPOTHÈSES DE BASE ET RESTRICTIONS

Les recherches ont été effectuées selon plusieurs hypothèses et restrictions, en voici la liste; [2] [3] [4]

- 1 - Une commande consiste en une séquence ordonnée d'opérations. Le cheminement est établi d'avance et doit être respecté. Chaque opération a au plus une opération la précédant et au plus une la succédant. L'assemblage n'est pas permis.
- 2 - Un type d'opération peut être exécuté par un seul type de machine.

- 3 - Il y a seulement une seule machine de chaque type dans l'atelier.
- 4 - Les temps d'exécution et les dates d'exigibilité sont connus à l'arrivée de la commande.
- 5 - Les temps de mise en route (set-up) sont indépendants de l'ordre d'exécution des commandes.
- 6 - Lorsqu'une opération débute sur une machine, elle ne doit pas être interrompue: pas de préemption.
- 7 - Une opération ne peut débiter avant que les opérations qui la précèdent dans la séquence soient terminées.
- 8 - Chaque machine est continuellement disponible. On ne tient pas compte des horaires de travail, des pauses, des bris etc.
- 9 - Chaque machine ne peut exécuter qu'une seule opération à la fois.
- 10 - Les inventaires de produits en cours sont permis.
- 11 - Le temps de transport entre les machines est considéré nul.

12 - On considère que les ressources humaines, les outils, fournitures etc. sont toujours disponibles et en quantités suffisantes.

D'autres études ont été effectuées en relaxant certaines de ces hypothèses. Cependant, cela augmente la complexité du problème. On décrira quelques unes de ces relaxations à la fin de cette section.

2.5 RÈGLES DE PRIORITÉ

Comme dit précédemment, la majorité des recherches ont comparé un certain nombre de règles de priorité. Par définition, une règle de priorité permet de choisir lorsqu'une machine se libère, quelle est la prochaine opération qui va être exécutée sur cette machine. Ce choix se fait parmi toutes les opérations pouvant être exécutée sur cette machine. Dans le cas particulier où les opérations sont déjà assignées aux machines, on regroupe les opérations en files d'attente devant chaque machine. Le choix se fait dans cette file. Voici maintenant une liste des règles les plus usuelles. Pour une liste plus complète consulter Blackstone, Phillips et Hogg [13] et Panwalkar et Iskander [14]. On choisira la commande:

- qui est arrivée la première à la machine (FCFS)
- qui est arrivée la première à l'atelier (FASFS)

- qui a le temps d'exécution le plus court (SPT)
- qui a le temps d'exécution le plus long (LDT)
- qui a le plus de temps d'exécution restant (temps d'exécution de toutes les opérations qui restent à faire) (MWKR)
- qui a le moins de temps d'exécution restant (LWKR)
- qui a le plus d'opérations restantes (MOPNR)
- qui a le moins d'opérations restantes (FOPNR)
- dont la somme des temps d'exécution des opérations en attente de la prochaine machine de la séquence est la plus petite (WINQ)
- dont la somme des temps d'exécution des opérations en attente de la prochaine machine de la séquence incluant les temps d'exécution des opérations qui sont prévues d'arriver est la plus petite (XWINQ)
- qui a la date d'exigibilité la plus tôt (DUEDATE)
- dont la différence entre la date d'exigibilité et la somme des temps d'exécution des opérations restantes est la plus petite (MST)
- dont la date d'exigibilité de l'opération est le plus tôt (OPNDD)
- dont la différence entre la date d'exigibilité et la somme des temps d'exécution des opérations restantes divisée par le nombre d'opérations restantes est le plus petit (S/OPN)
- qui a le temps d'exécution le plus court à moins qu'une commande ait été retenue dans la queue plus qu'une certaine limite de temps (TSPT)

- qui est arrivée la première à la machine à moins que la longueur de la queue dépasse une certaine limite, alors ce sera le temps d'exécution le plus court (RSPT)
- au hasard (RANDOM)

Cette liste n'est pas exhaustive. En réalité, il existe une infinité de règles; on peut faire des ratios avec certaines, utiliser plusieurs règles à la fois en les pondérant, subdiviser les commandes en plusieurs classes, etc. De plus certaines règles sont beaucoup plus simple que d'autres à implanter. Il faut donc faire le rapport coûts/bénéfices avant de choisir une règle.

2.6 RÉSULTATS DES RECHERCHES

La majorité des résultats obtenus sur l'atelier de sous-traitance général découlent des expériences de Conway, Maxwell et Miller [3]. Ils ont simulé un atelier de 9 machines avec environ 10 000 commandes et comparé plusieurs règles de priorité en fonction de diverses mesures de performance. Voyons maintenant les résultats obtenus en fonction de deux mesures de performance couramment utilisées.

Réduction du temps moyen en atelier

Il a été démontré que minimiser le temps moyen en atelier est équivalent à minimiser le nombre moyen de commandes en atelier [2] [3]. Voici un tableau sommaire des résultats de Conway.

Règles de priorité	Nombre moyen de commandes dans l'atelier
SPT	23,25
LWKR	47,25
FOPNR	52,23
WINQ	40,43
XWINQ	34,03
RANDOM	59,42
FCFS	58,87
.97 SPT + 0,03 WINQ	22,83
.96 SPT + 0,04 XWINQ	22,67
.985 SPT + 0,015 LWKR	22,98

Tableau 2.1

Résultats de certaines règles en fonction du nombre moyen de commandes dans l'atelier

La règle SPT domine largement les autres règles simples. Conway a aussi étudié la possibilité de combiner certaines règles. Certaines améliorent légèrement les résultats, cependant le choix de la pondération est extrêmement difficile à faire et est très sensible aux variations de paramètres tel le pourcentage d'utilisation de l'atelier et de plus certaines règles comme WINQ et XWINQ sont difficiles à implanter. L'effort requis n'est donc pas justifiable pour cette légère amélioration.

Les études de Nanot [15] ont démontré que la règle SPT dominait constamment les autres lorsqu'on désire minimiser le temps moyen en atelier.

Cependant un problème qui peut subvenir en utilisant la règle SPT est qu'une commande ayant des temps d'exécution très longs reste prise dans l'atelier et prend longtemps à s'en sortir. Conway a donc étudié les règles TSPT et RSPT à ce sujet. Les résultats sont;

Règles de priorité		nb moyen de commandes dans l'atelier
TSPT attente max =	∞	23,25
TSPT	32	32,85
TSPT	16	44,20
TSPT	8	53,50
TSPT	4	55,67
TSPT	0	58,87
RSPT queue max =	1	23,25
RSPT	5	29,49
RSPT	9	38,67
RSPT	∞	58,87

Tableau 2.2

Résultats de TSPT et RSPT en fonction du nombre moyen de commandes dans l'atelier

Des règles telles TSPT et RSPT doivent être utilisées si l'atelier ne permet pas qu'une commande ait un temps en atelier excessif.

Respect des dates d'exigibilité

Pour certains ateliers, le respect des dates d'exigibilité est prépondérant. Dans ce cas, il semble à priori préférable d'utiliser des règles de priorité qui tiennent compte de ces dates. Conway a obtenu les résultats suivants pour quelques règles.

règles de priorité	écart exigibilité terminaison moyen	variance	% de commandes en retard
RANDOM	-4,2	6914	30,75
FCFS	-4,5	1686	44,79
DUEDATE	-15,5	432	17,75
S/OPN	-12,8	226	3,71
OPNDD	-9,9	14560	10,36
SPT	-44,9	2878	5,02

Tableau 2.3

Résultats de quelques règles en fonction de critères ayant rapport aux dates d'exigibilité

NOTE: Ces résultats ont été obtenus selon un taux d'utilisation de 88,4% et avec les dates d'exigibilité fixées proportionnellement au temps total d'exécution requis.

Les résultats varient selon la méthode utilisée pour fixer les dates d'exigibilité et en fonction du taux d'utilisation de l'atelier. Les expériences de Conway ont démontré que la règle SPT est beaucoup moins sensible que les autres et qu'elle donne constamment de bon résultats, ce qui semble surprenant car la règle SPT ne

tient pas compte intrinsèquement de la date d'exigibilité.

En résumé, la règle SPT offre toujours de bons résultats sauf en ce qui a trait à la variance de l'écart exigibilité-terminaison. Si l'on désire diminuer cette variance, on doit faire appel à une règle utilisant la date d'exigibilité.

2.7 RELAXATION DE CERTAINES HYPOTHÈSES ET RESTRICTIONS

Certaines études ont été réalisées en relaxant certaines hypothèses et restrictions de base décrites précédemment soit pour augmenter le degré de réalisme ou pour étudier un cas plus spécifique. Cependant, éliminer une de ces hypothèses augmente la complexité du problème, c'est pourquoi la majorité de ces études ont surtout été faites pour des cas dégénérés.

On présente ici sommairement quelques sujets sur lesquelles certains chercheurs se sont attaqués en n'oubliant pas que nous étudions l'atelier de sous-traitance général dynamique stochastique.

. Assemblage

On permet ici qu'une opération ait plus d'une opération la précédant, consulter [3], [16].

. Plusieurs cheminement possibles

On considère que la séquence d'opérations nécessaires à la réalisation d'un produit peut être en partie flexible, consulter [3], [17].

. Plusieurs machines pouvant exécuter un même type d'opération

On considère le cas où une opération peut être effectuée sur une autre machine que celle prévue en cas de congestion, consulter [3], [18].

2.8 CONCLUSION

L'ordonnancement d'un atelier de sous-traitance général (i.e. plusieurs machines) est une tâche très difficile et on ne peut espérer obtenir une solution optimale vue la complexité du problème. Cependant, étant une tâche nécessaire au bon fonctionnement d'une entreprise, de nombreux chercheurs se sont attaqués à ce problème. Les chercheurs ont démontré que la règle de temps de procédé le plus court (SPT) était celle qui offrait constamment des bons résultats que ce soit pour le temps en atelier ou plus le respect des dates d'exigibilité et qu'elle devrait être considérée comme standard selon lequel les autres règles devront faire leurs preuves.

Malheureusement, ces études ont été réalisées selon une liste d'hypothèses et restrictions qui réduisent leur applicabilité. Dans la prochaine section, on éliminera une de ces contraintes pour considérer le cas où une machine peut traiter plusieurs pièces en même temps.

CHAPITRE 3

LA PROBLÉMATIQUE DE LA MACHINE À TRAITEMENT PAR LOT

3.1 INTRODUCTION

Le but de cette recherche est d'étudier le problème d'ordonnancement de production dans un atelier de sous-traitance. Cependant, contrairement aux études antérieures qui supposaient qu'une machine ne peut qu'exécuter qu'une seule pièce à la fois, nous considérons le cas plus général où une machine peut traiter plusieurs pièces simultanément. On appellera une telle machine une machine à traitement par lot. On tentera de trouver une méthode la plus générique possible pouvant être applicable aux divers types de machines à traitement par lot que l'on retrouve dans l'industrie étant donné que les facteurs qui influencent la sélection des pièces formant un lot peuvent varier considérablement.

3.2 TYPES DE PROCÉDÉS

On retrouve dans l'industrie plusieurs procédés qui utilisent ce type de machine. Les plus connus sont le traitement thermique qui à pour but de donner

aux pièces certaines caractéristiques comme une dureté spécifique, les équipements de nettoyage (deburring machines) qui permettent d'améliorer considérablement la finition des produits finis et finalement divers procédés chimiques tels des bains pour recouvrir les pièces d'un enduit anti-corrosif, un enduit final telle une peinture ou autre. D'autres procédés moins fréquents peuvent aussi utilisés ce type de machine. On s'aperçoit facilement que ces diverses machines à traitement par lot, même si elles peuvent traiter plusieurs pièces simultanément peuvent avoir certaines exigences quant aux pièces pouvant être traitées en même temps. On introduit ici les familles de pièces. Une famille étant un groupe de pièces pouvant être traitées simultanément. Voici une liste sommaire de différents facteurs qui peuvent entrer en jeu lors de la détermination des familles de pièces;

- type de matériel
- similarité de poids, volume, surface, longueur
- temps de procédé
- type de traitement

3.3 INTÉRÊT DE L'ÉTUDE

L'intérêt d'étudier la machine à traitement par lot vient surtout du fait que les temps d'exécution de cette machine peuvent être beaucoup plus longs que pour une machine ordinaire (ex: un traitement thermique peut exiger facilement

plusieurs heures de préchauffage, de traitement et de refroidissement). Ce type de machine peut représenter un goulot d'étranglement dans l'atelier et/ou peut être la machine la plus coûteuse à faire fonctionner. De plus, en consultant la littérature sur l'ordonnancement de production, il semble que personne ne s'est attaqué à ce problème.

Cette recherche aura pour but d'élaborer une méthode qui nous permettra d'ordonner la production dans un atelier de sous-traitance comprenant une machine à traitement par lot en plus de plusieurs machines ordinaires (qui traitent une seule pièce à la fois). Les dernières sont plus simples à ordonner car de nombreuses règles de priorité existent pour ces machines. Quant à la machine à traitement par lot, il faudra probablement lui ajouter des règles propres étant donné son mode de fonctionnement particulier.

3.4 ATELIER TYPE

Pour étudier comment ordonner une machine à traitement par lot, on simule un petit atelier de sous-traitance fictif comprenant 9 machines ordinaires et une machine à traitement par lot. Il a été en effet montré que les résultats obtenus sur l'ordonnancement d'un petit atelier sont souvent généralisables pour des gros ateliers comprenant plus de cent machines [3]. Notre atelier fonctionnera en mode dynamique (les commandes arrivent aléatoirement dans le temps et les

temps d'arrivée ne sont pas connus d'avance. Les cheminements et temps d'exécution des pièces dans l'atelier sont aussi aléatoires mais sont par contre déterminés à l'arrivée d'une commande à l'atelier. Notre atelier sera donc comme celui ci-dessous:

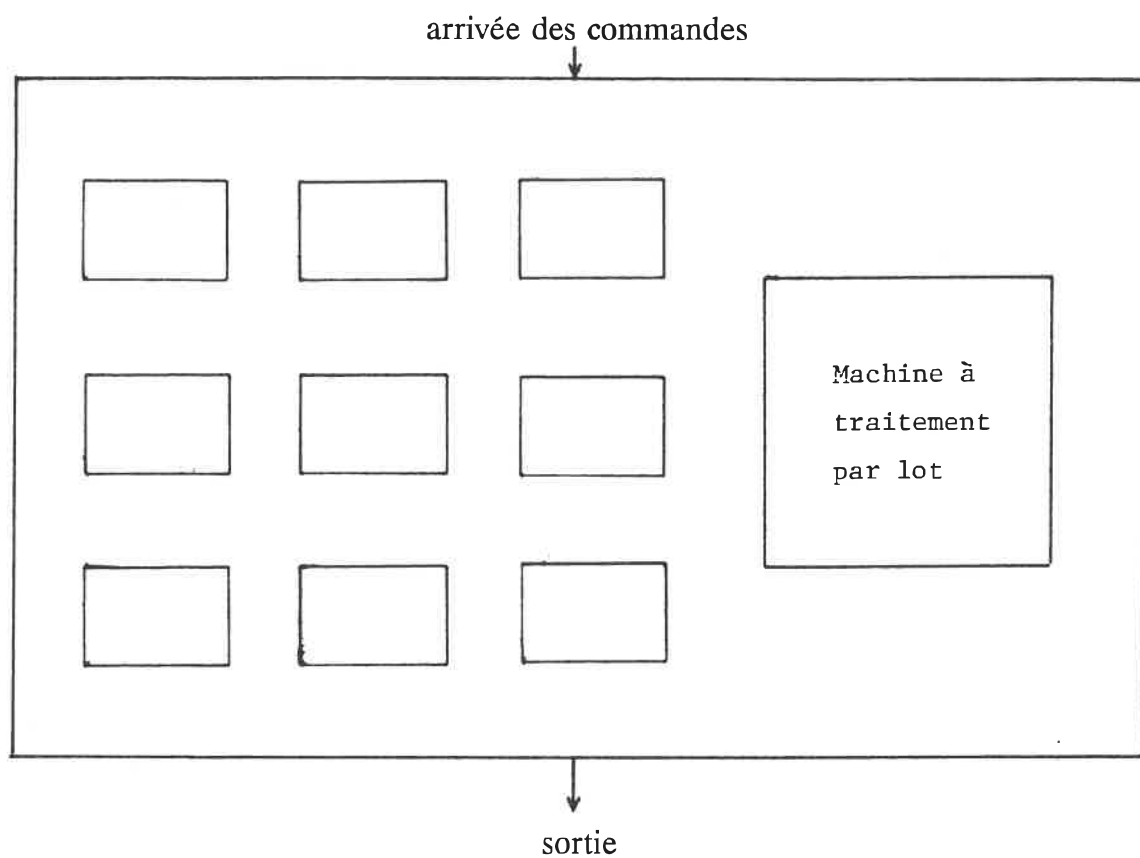


Fig 3.1

Atelier de sous-traitance

3.5 HYPOTHÈSES ET RESTRICTIONS DE BASE

Comme pour les recherches antérieures, diverses hypothèses sont nécessaires pour bien définir notre problème. Elles sont:

- une commande consiste en une série ordonnée d'opérations. Il n'y a pas d'assemblage
- il y a seulement une seule machine de chaque type et un type d'opération ne peut être exécuté que par un type de machine
- Les temps d'exécution, les dates d'exigibilité, l'ordre par lequel les machines sont visitées ainsi que d'autres paramètres sont déterminés lorsque la commande arrive à l'atelier et restent constants
- les temps de mise en route sont indépendants de l'ordre d'exécution des commandes
- pas d'interruption des opérations en cours
- une opération ne peut débuter avant que les opérations qui la précèdent dans la séquence soient terminées

- on ne tient pas compte des bris, horaires de travail, pauses etc.
- les inventaires de produits en cours sont permis
- les temps de transport entre deux machines sont considérés constants. On les ajoute au temps de traitement
- les ressources humaines, outils, fournitures etc. sont toujours disponibles et en quantité suffisante

En fait la seule différence par rapport aux études antérieures et que dans notre cas on permet qu'une machine exécute plusieurs pièces en même temps.

3.6 ORDONNANCEMENT DES MACHINES ORDINAIRES

Étant donné qu'on s'intéresse plus particulièrement à l'ordonnancement de la machine à traitement par lot, on effectuera l'ordonnancement des machines ordinaires selon la règle de temps d'exécution le plus court (SPT): cette règle offre de bons résultats pour plusieurs mesures de performance et elle est facile à implanter. De plus en conservant toujours la même règle pour les machines ordinaires, il nous sera plus facile de comparer les diverses méthodes ou règles utilisées pour ordonnancer les tâches sur la machine à traitement par lot.

Ultérieurement on pourrait s'interroger sur l'efficacité de la règle SPT pour les machines ordinaires lorsqu'on ajoute une machine à traitement par lot à l'atelier. Cependant, cela ne fera pas objet de ce travail.

3.7 FACTEURS DISCRIMINANTS POUR LA MACHINE À TRAITEMENT PAR LOT

Certaines caractéristiques de la machine à traitement par lot et de l'atelier où elle est utilisée peuvent avoir un impact assez important sur l'ordonnancement de la production. En voici la liste:

- Situation de la machine à traitement par lot dans le processus (amont ou aval).
Dépendant du type d'atelier ou du type de machine à traitement par lot, les cheminements des diverses commandes peuvent inclure cette machine dans les premières machines visitées ou contrairement dans les dernières.
- Importance de la machine à traitement par lot par rapport aux machines ordinaires. La machine à traitement par lot représente-t-elle un goulot d'étranglement ?

- La machine à traitement par lot peut avoir un certain coût de fonctionnement. Il peut donc devenir indésirable de la faire fonctionner à faible capacité. Il peut être préférable d'attendre d'avoir plus de pièces avant de démarrer cette machine. Il faut donc faire une relation entre le coût de fonctionner à faible capacité et le coût d'attente.
- Temps d'exécution de la machine à traitement par lot par rapport aux machines ordinaires.
- Nombre de familles pièces. Étant donné que la machine à traitement par lot traite plusieurs pièces en même temps, il faut souvent un certain degré de similitude entre les pièces traitées en même temps. Les principaux facteurs déterminant les familles ont été mentionnés précédemment.
- Pièces avec temps d'attente maximum. Si certaines pièces après avoir été exécutées par la machine ordinaire précédant la machine à traitement par lot dans la séquence ne peuvent attendre plus qu'un certain temps avant d'être exécutées par cette dernière.
- Le temps de mise en route peut être dépendant de la séquence par laquelle les familles sont traitées. Il y a aussi le cas limite où le passage d'une famille à une autre est techniquement impossible.

Cette liste regroupe les principaux facteurs et cas qui peuvent subvenir dans l'industrie en ce qui a trait à la machine à traitement par lot. Cependant notre étude se concentrera beaucoup plus sur les premiers points étant donné que pour les deux derniers, il semble à priori qu'une planification globale de production sont préférable. On se contentera de prendre des décisions d'ordre tactique lors de cette étude.

3.8 CLASSIFICATION DES FACTEURS

Après avoir dressé la liste des principaux facteurs discriminants pour la machine à traitement par lot, il a été possible de les classer en deux catégories distinctes; contraintes de fonctionnement imposées par les produits ou par la machine à traitement par lot et la prépondérance de cette machine. On ne classe pas ici les deux derniers facteurs mentionnés précédemment puisqu'ils ne seront pas analysés. Les facteurs catégorisant la première classe sont:

- Contraintes de fonctionnement imposées par les produits ou par la machine à traitement par lot.
 - . Nombre de familles de pièces
 - . Coût de fonctionnement versus coût d'attente

Le nombre de familles de pièces varie en fonction des caractéristiques de la machine ou des produits. Si les tolérances ou caractéristiques des produits sont très exigeantes, il peut être difficile d'exécuter des pièces différentes en même temps. Si les coûts d'utilisation ou les temps d'exécution sont assez long, il peut être désavantageux de faire fonctionner la machine à traitement par lot à faible capacité.

Les facteurs catégorisant la deuxième classe sont:

- Prépondérance de la machine à traitement par lot.
 - . Situation dans le processus
 - . Importance de cette machine par rapport aux machines ordinaires
 - . Temps d'exécution de cette machine par rapport aux machines ordinaires

L'exemple le plus évident pour la prépondérance est le four à traitement thermique. Il peut se situer plus vers le début ou plus vers la fin dans le cheminement tout dépendant du type de procédé que l'on désire. De plus les temps d'exécution sont généralement longs car les traitements thermiques exigent fréquemment des taux de chauffage et de refroidissement lents pour donner aux pièces les caractéristiques désirées.

3.9 DÉCISIONS À PRENDRE LORS DE L'ORDONNANCEMENT

. Cas classique

Dans le cas où nous avons que des machines ordinaires, le problème d'ordonnancement de production est beaucoup plus simple. Lorsqu'une machine se libère, une décision se prend à savoir laquelle parmi les pièces en attente devant cette machine va être la prochaine à être traitée. Pour choisir une pièce, on utilise une de nombreuses règles de priorité qui nous permettra de viser certains objectifs de performance pour l'atelier. Les décisions sont du point de vue tactique seulement et l'atelier est représenté par un réseau de queues où les commandes circulent depuis leur arrivée jusqu'à leur terminaison.

. Avec une machine à traitement par lot

En ajoutant une machine à traitement par lot à l'atelier, la prise de décision n'est plus simpliste pour cette machine. On retrouve 3 décisions de niveau tactique à prendre. Elles sont:

1 - Quelle famille choisir ?

Après avoir traité des pièces provenant d'une famille, de quelle famille va-t-on exécuter le prochain lot ? Pour ce faire, on devra analyser quelles sont les

pièces en attente, le nombre ainsi que les paramètres de celles-ci.

2 - Parmi la famille choisie, quelle pièces formeront le lot ?

On considérera ici la capacité de la machine ainsi que les paramètres des pièces en attente.

3 - Est-il préférable d'attendre avant de démarrer la machine si elle n'est pas remplie à pleine capacité ?

Dans le cas où le temps d'exécution est assez long, il peut être coûteux de faire fonctionner la machine à traitement par lot à faible capacité.

3.10 OBJET DE L'ÉTUDE

Dans cette étude, on tentera de répondre aux trois questions de niveau tactique et de trouver les meilleures règles de priorité ou procédures pour la machine à traitement par lot en vue d'améliorer l'efficacité globale de l'atelier.

Il pourrait être possible que des décisions d'ordre stratégique améliore le rendement de la machine à traitement par lot et donc de l'atelier. Cependant cela requiert une différente approche du problème et pourrait être utile pour des cas plus spécifique. Nous nous concentrerons sur les 3 décisions d'ordre tactique.

CHAPITRE 4

ORDONNANCEMENT DE LA MACHINE À TRAITEMENT PAR LOT

4.1 INTRODUCTION

Tel que mentionné dans la section précédente, le problème d'ordonnancement de production dans un atelier de sous-traitance comportant une machine à traitement par lot est plus complexe. Les décisions du point de vue tactique sont moins directes que pour une machine ordinaire. On se doit de prendre des décisions sur les trois questions à savoir quelle famille choisir, quelle pièces dans cette famille et s'il est préférable d'attendre avant de faire partir la machine à traitement par lot. Ces trois décisions sont cruciales et ont un impact direct sur les performances de l'atelier.

4.2 SCHÉMATISATION DE LA MACHINE À TRAITEMENT PAR LOT

Les études antérieures qui ont analysé l'atelier de sous-traitance général considéraient l'atelier comme un réseau de queues (une par machine).

Voir figure.

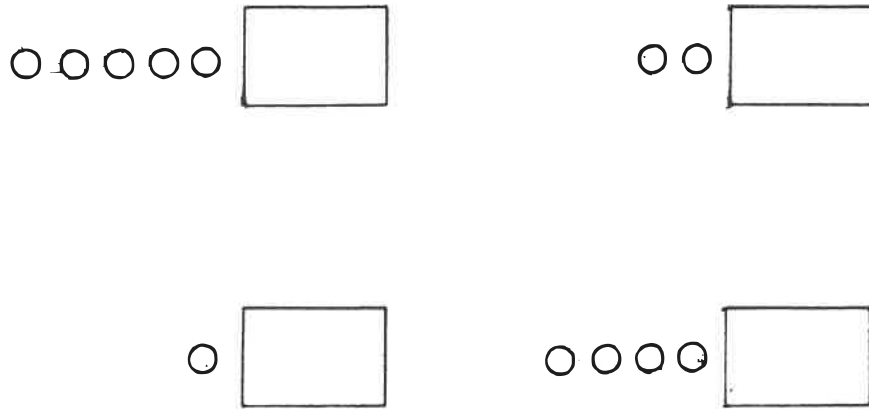


Fig. 4.1

Atelier de sous-traitance général avec machines ordinaires seulement. Lorsqu'une machine se libère, le choix se fait parmi les pièces en attente

On pourrait schématiser notre machine à traitement par lot par la figure qui suit:

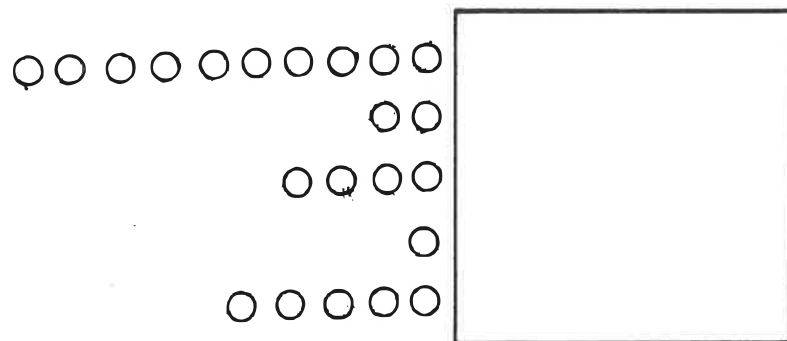


Fig. 4.2

Machine à traitement par lot

On aura une file d'attente par famille, des pièces en attente ou non pour chacune des familles et une capacité pour la machine. Cette capacité pourrait tout dépendant du type de machine être en terme de poids, surface, nombre, etc. Pour les besoins de notre étude, on fixera une capacité maximum de 10 pièces pouvant être traitées simultanément. On considère aussi qu'une pièce ne peut faire partie d'une seule famille et qu'un maximum de 5 familles peuvent exister pour cette machine. La raison pour laquelle on restreint le nombre de familles est pour nous assurer d'être capable de former des lots de grosseur raisonnable et non pas d'avoir une multitude de files d'attente comprenant seulement une, deux ou même aucune pièce en attente.

4.3 RÈGLES DE PRIORITÉ ÉTUDIÉES

Comme mentionné antérieurement, au point de vue tactique nous avons trois questions à résoudre. Comme pour les recherches qui ont été faites sur l'ordonnancement de production, on utilisera autant que possible des règles de priorité pour y répondre. Certaines de ces règles sont fréquemment utilisées dans la littérature, d'autres sont nouvelles étant donné les caractéristiques différentes de la machine à traitement par lot par rapport aux machines ordinaires.

On se limite aussi à n'étudier que quelques règles pour chacune des questions. Cela pour éviter d'avoir un nombre trop grand de combinaisons de règles. Les

règles sont pour chacune des questions:

1 - Quelle famille choisir ?

- . SPT: Choisir la famille qui a le temps de procédé le plus court.

- . WORKLOAD: Choisir la famille qui a le plus de pièces en attente.

- . RANDOM: au hasard

Ces 3 règles sont simples et faciles à implanter. D'autres règles qui tiennent compte des caractéristiques de chacune des pièces formant le lot sont envisageables. Par contre elles sont plus compliquées. Cette complication vient du fait que les pièces en attente pour une famille ne sont pas automatiquement celles qui formeraient le lot à être exécuté. Pour ce type de règles, il faudrait répondre premièrement à la question 2 pour savoir quelles pièces formeraient le lot si la famille était choisie. On analysera une règle de ce type.

- . AVGDUEDATE: Choisir la famille dont la moyenne des dates d'exigibilité est la plus petite (pour éliminer le plus de retards possibles)

2 - Dans une famille, quelles pièces choisir pour former le lot ?

Cette décision est nécessaire seulement si le nombre de pièces en attente est plus grand que la capacité de la machine à traitement par lot. Les règles étudiées sont:

- . DUE DATE: Choisir les pièces dont les dates d'exigibilité sont les plus rapprochées.
- . SLACK: Cette méthode est une sorte de date d'exigibilité révisée. On définit le "SLACK" comme étant:

$$\text{SLACK} = \frac{\text{Date d'exigibilité} - \text{somme des temps des opérations restantes} - \text{temps actuel}}{\text{Somme des temps des opérations restantes}}$$

En effet, si une pièce a pris beaucoup de temps lors de l'exécution des opérations sur les machines précédant la machine à traitement par lot, le SLACK sera petit et forcera cette pièce à passer avant les autres et vice-versa.

- . RANDOM: Au hasard.

3 - Est-il préférable d'attendre si la machine à traitement par lot n'est pas à pleine capacité?

On devra faire un compromis entre le coût d'attente (machine à traitement par lot libre) et le gain en production provenant d'une meilleure utilisation de la capacité de cette machine. L'attente donne du retard supplémentaire aux pièces mais par contre le coût unitaire d'utilisation diminue si la machine à traitement par lot fonctionne à plus grande capacité.

Cette question est plus difficile que les autres à répondre car on se doit de faire des prévisions sur les pièces qui vont arriver à la machine à traitement par lot dans un avenir rapproché. On utilisera pour ce faire le taux moyen d'arrivée des pièces par famille à cette machine étant donné que notre système va fonctionner en régime permanent et sera assez stable. Ce taux est le même pour chaque famille. La question est de savoir si la productivité de la machine à traitement par lot augmente en fonction du temps d'attente. Pour ce faire, nous avons élaboré 2 formules de productivité en fonction du temps, elles sont:

$$\text{Prod. 1} = \frac{\text{Nombre de pièces présentes + arrivées en fonction du temps d'attente}}{\text{Temps d'exécution + temps d'attente + coût fixe}}$$

La particularité de cette formule est qu'on y met un coût fixe au dénominateur. Ce coût peut représenter le coût de démarrer la machine. Il peut donc être préférable dans ce cas-ci de répartir ce coût sur plusieurs pièces. La détermination du coût fixe est libre à l'utilisateur.

$$\text{Prod. 2} = \frac{\text{Nombre de pièces présentes + arrivées en fonction du temps d'attente}}{\alpha \text{ temps d'exécution} + \text{temps d'attente}} \times \text{Nombre de pièces présentes + arrivées en fonction du temps}$$

Cette méthode met plus d'emphasis sur le temps d'exécution d'une famille quelconque. En effet il semble à priori que plus le temps sera long, plus on exigera un grand nombre de pièces avant de démarrer la machine à traitement par lot. Le choix de α revient aussi à l'utilisateur.

Peut importe la méthode utilisée, on calculera la productivité de la machine en fonction du temps d'attente. Cela pourra nous donner des courbes comme sur la figure suivante:

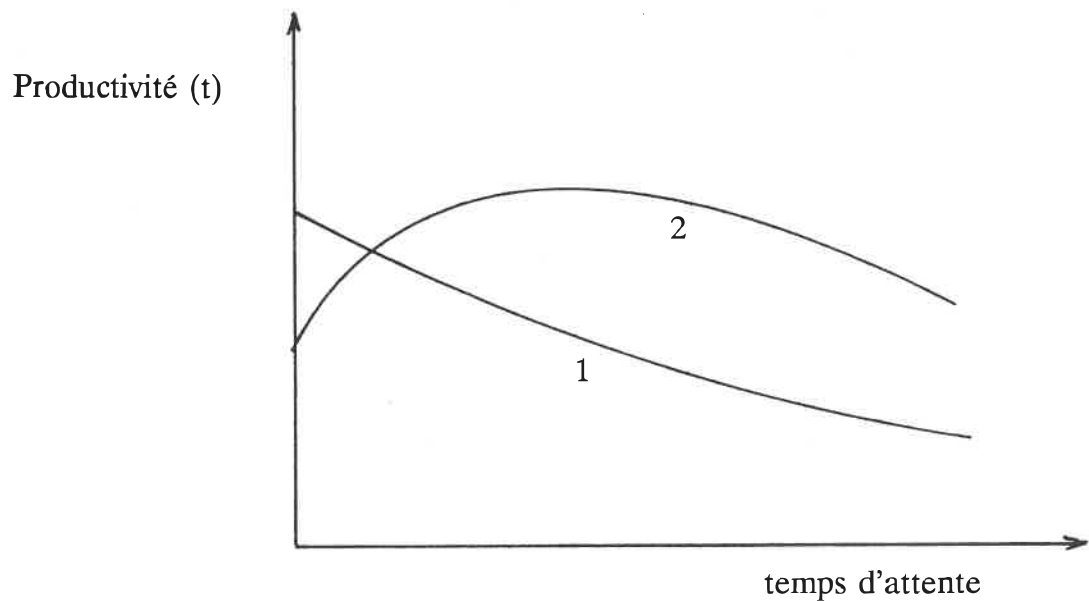


Fig. 4.3

Productivité en fonction du temps d'attente

On démarrera la machine à traitement par lot si la valeur maximale se retrouve à l'ordonnée. Dans ce cas, la productivité de la machine est maximale si nous n'attendons pas. Par exemple, dans le cas 1 de la figure précédente, le départ sera immédiat. Dans le cas 2, il est préférable d'attendre car la productivité augmente si on attend une certaine période de temps.

Dans le cas où il est préférable d'attendre, on bloquera le départ de la machine à traitement par lot pour un petit intervalle de temps t . On recalculera par la suite les valeurs de productivité en fonction du temps. Si la valeur maximale se retrouve à l'ordonnée, on démarrera la machine, sinon on bloquera encore une fois le départ. On répétera cette procédure jusqu'à temps que la valeur maximale se retrouve à l'ordonnée. En dernier lieu, il faut remarquer que peut importe la formule de productivité utilisée, la somme des termes nombre de pièces présentes et arrivées en fonction du temps ne pourra jamais dépasser 10 (capacité de la machine) quelque soit le temps d'attente.

Il est bien sûr logique de penser que d'autres formules de productivité sont envisageables. Par exemple, on aurait pu ajouter un poids au temps d'attente du dénominateur due à l'incertitude de l'arrivée des pièces étant donné les phénomènes aléatoires qui peuvent se produire.

4.4 PRISE DE DÉCISIONS

Avant de mettre un lot de pièces dans la machine à traitement par lot et de la faire démarrer, il faut répondre aux trois questions mentionnées précédemment. Une difficulté qui survient est que les réponses sont interdépendantes. Dépendant de la situation dans laquelle on se retrouve et des règles de priorité étudiées, on peut-être appelé à répondre dans l'ordre 1-2-3 ou 2-1-3. Par exemple si on utilise la règle AVGDUE DATE pour choisir la famille, il est nécessaire de venir former le lot optimum pour chaque famille avant. Si par contre on utilise SPT, il n'est pas nécessaire de répondre à la question 2 en premier mais en réalité tous les cas peuvent être résolus selon l'ordre 2-1-3 même si parfois il serait plus simple ou direct d'utiliser l'ordre 1-2-3. Pour faciliter la programmation, il est préférable d'utiliser une méthode rigide s'appliquant à tous les cas. Les étapes sont les suivants:

Étape 1: Question 2

Pour chaque famille de pièces, former le lot optimum en utilisant une des règles de priorité. Pour chacun de ces lots, on lui attribue une valeur actuelle représentant son contenu.

Étape 2: Question 1

Choisir la meilleure famille en tenant compte des règles qui s'appliquent.

Étape 3: Question 3

Si le lot choisi peut remplir la machine à traitement par lot à pleine capacité, on n'a pas à considérer l'attente. À l'opposé, on estimera la productivité en fonction du temps pour la famille choisie pour déterminer si une attente pourrait être préférable.

CHAPITRE 5

PLAN D'EXPÉRIENCE

5.1 INTRODUCTION

Le problème d'ordonnancement avec une machine à traitement par lot nécessite plusieurs types de décisions: quelle famille de pièces choisir, quelles pièces parmi cette famille et exécuter ou attendre. Il nous semble donc préférable à priori de fixer certains paramètres et de limiter le nombre de combinaisons de règles à étudier. Cela nous permettra d'analyser plus en profondeur les performances de celles étudiées. De plus on pourrait envisager que les résultats obtenus pourront être généralisés à des cas similaires ayant des paramètres de base légèrement différents. Notre étude sera divisée en trois étapes majeures présentées en 5.2, 5.3 et 5.4. La section 5.5 présente les jeux de données utilisées par chacune des études et la section 5.6, les résultats produits à chaque simulation.

5.2 CHOIX DE LA FAMILLE ET DES PIÈCES SEULEMENT

Dans cette première étape, on ne s'occupera pas de savoir si une attente pourrait être préférable. On cherchera plutôt à obtenir des résultats de base qui nous permettront plus tard d'analyser les résultats des autres alternatives. Les combinaisons de règles de priorité étudiées sont les suivantes (la première est la règle qui permet de choisir les pièces parmi la famille et la seconde permet de choisir la famille):

- . **RANDOM-RANDOM:** Le choix des pièces et de la famille se font au hasard. On ne s'attend évidemment pas à de bons résultats avec cette combinaison. Elle nous servira de base de comparaison pour les autres combinaisons qui suivent.

- . **DUEDATE-SPT:** On choisira la famille avec le temps d'exécution le plus court et choisirons les pièces qui ont les dates d'exigibilité les plus rapprochées. À priori cette règle semble pouvoir offrir de bons résultats. Cette combinaison semble avoir l'avantage de vouloir réduire le temps en atelier grâce à SPT et à respecter les dates d'exigibilité grâce à DUEDATE. Ce sont les deux critères qui ont été

le plus fréquemment utilisés lors des recherches antérieures.

- . **DUEDATE-AVGDUEDATE:** On choisit la famille dont la moyenne des dates d'exigibilité est la plus petite et on choisit dans cette famille les pièces qui ont les dates d'exigibilité les plus rapprochées.

- . **DUEDATE-WORKLOAD:** On choisira la famille qui a le plus de pièces en attente. On choisira ensuite les pièces selon les dates d'exigibilité. Cette combinaison met l'emphase sur la productivité de la machine à traitement par lot. Elle devrait offrir des bons résultats car elle utilise au maximum la machine à traitement par lot, libérera le plus de pièces possibles aux autres machines de l'atelier, ce qui permettra éventuellement à ces pièces de quitter l'atelier dans de bons délais.

- . **SLACK-WORKLOAD:** On choisira la famille qui a le plus de pièces en attente. On choisira ensuite les pièces selon le SLACK. Cette règle offre les mêmes avantages que la précédente. Cependant le SLACK devrait

améliorer le respect des dates d'exigibilité car il est en effet une sorte de date d'exigibilité révisée en fonction du temps. Le désavantage de cette règle par rapport à la précédente est qu'elle est plus difficile à implanter et qu'elle pourra demander plus de temps d'exécution de la part de l'ordinateur.

5.3 POSITION DE LA MACHINE À TRAITEMENT PAR LOT

Dans ce cas-ci, on étudiera si le positionnement de la machine peut avoir un effet sur les performances de l'atelier. Pour ce faire on placera la machine vers la fin du cheminement de chacune des pièces. En plaçant la machine ainsi, on la rend plus critique car le choix de la famille et des pièces peuvent avoir un effet plus direct sur le respect des dates d'exigibilité. Les 2 décisions semblent devenir plus importantes.

Pour analyser l'effet d'un changement de positionnement sur les mesures de performances, on utilisera les 2 combinaisons de règles de priorité qui ont obtenu les meilleurs résultats dans la section précédente.

5.4 POLITIQUE D'ATTENTE POUR LA MACHINE À TRAITEMENT PAR LOT

L'objectif de cette analyse est de voir l'effet sur les performances globales de l'atelier d'avoir comme sous-objectif de maximiser la productivité de la machine à traitement par lot. Ce que l'on désire ici c'est d'avoir des lots de pièces de tailles importantes étant donné que pour certains types d'atelier le coût de fonctionnement de cette machine peut être élevé. À priori, il semble que les temps en atelier pourraient augmenter et les dates d'exigibilité être moins bien respectées étant donné les temps morts qu'il pourrait y avoir à cette machine.

Pour analyser l'effet de l'attente, on utilisera encore une fois les 2 meilleures combinaisons de règles de priorité obtenues dans la section 5.2.

5.5 PARAMÈTRES DES PIÈCES À EXÉCUTER

Compte tenu du nombre de combinaisons à simuler pour obtenir nos résultats, il est préférable de fixer d'avance certains paramètres pour conserver un nombre raisonnable d'exécutions.

Le paramètre le plus important à faire varier lors de ces exécutions est le temps moyen de la machine à traitement par lot par rapports aux machines ordinaires. On aura 6 temps différents qui auront pour but de rendre cette machine

plus ou moins critique. On analysera les résultats des différentes combinaisons et vérifierons leur efficacité en fonction du temps de traitement de la machine à traitement par lot.

On fixera le nombre de familles de pièces à 5. Ce nombre nous apparaît raisonnable car il offre une assez grande diversité et nous permet de s'assurer que des lots de grosseur acceptable pourront être formés. De plus, le temps de traitement de chacune des familles sera différent et fixé en fonction du temps moyen de la machine à traitement par lot. Les temps seront de 50%, 75%, 100%, 125% et 150% de ce temps moyen.

Quant aux machines ordinaires, on assignera aux pièces un temps d'exécution provenant d'une distribution exponentielle avec moyenne de 1. La raison du choix de cette distribution est qu'elle est celle qui a été utilisée lors des recherches antérieures. De plus on s'assurera d'avoir un bon taux de congestion dans l'atelier, tournant autour de 90% pour la machines ordinaires comme cela a été le cas pour la majorité des recherches faites sur le sujet. Comme dit précédemment ces machines seront toutes ordonnancées selon la règle SPT, ce qui nous donnera une bonne base pour comparer nos diverses combinaisons.

Finalement, les dates d'exigibilité des pièces seront assignées de façon proportionnelle à la somme de leur temps d'exécution sur les 10 machines.

5.6 STATISTIQUES

Plusieurs statistiques seront compilées lors des exécutions. Les plus importantes sont:

- Temps en atelier global et pour chacune des familles.
- Écart exigibilité-terminaison global et pour chacune des familles.
- Grosseur des lots pour la machine à traitement par lot pour chacune des familles.
- Temps moyen d'attente des pièces à la machine à traitement par lot pour chacune des familles.
- Nombre de lots exécutés par la machine à traitement par lot pour chacune des familles.
- Nombre de pièces en retard et en avance (par rapport à la date d'exigibilité) pour chacune des familles.
- Utilisation de la machine à traitement par lot.

L'analyse de ces statistiques nous permettra de déterminer la méthode d'ordonnement la plus efficace en ce qui à trait à la machine à traitement par lot.

CHAPITRE 6

PROGRAMMATION

6.1 INTRODUCTION

Notre problème est de trouver une méthode d'ordonnancement d'une machine à traitement par lot dans un atelier de sous-traitance. On sait d'avance qu'on ne peut trouver une solution optimale à ce problème étant donné que nous sommes dans un environnement dynamique. On veut plutôt étudier la moyenne à long terme des performances du système pour différentes politiques d'ordonnancement. Comme pour les études antérieures nous avons choisi d'utiliser un langage de simulation pour obtenir les résultats. Les prochaines sections décrivent les principaux aspects de la programmation.

6.2 CHOIX DU LANGAGE

Notre atelier a été programmé avec le langage SLAM II version 3.0 [19]. L'avantage de ce langage est qu'il est très flexible et permet une grande liberté au point de vue programmation. L'atelier sera représenté en utilisant les énoncés du

réseau SLAM et avec des sous-programmes Fortran qui soit améliorent l'efficacité de la programmation soit son degré de flexibilité. Consulter l'annexe pour les énoncés de programmation (pour la combinaison SLACK-WORKLOAD) et les dessins inclus en pochette pour le réseau. Le réseau représente le cheminement des entités (dans notre cas les pièces) dans l'atelier. On peut représenter la logique de programmation par la figure qui suit:

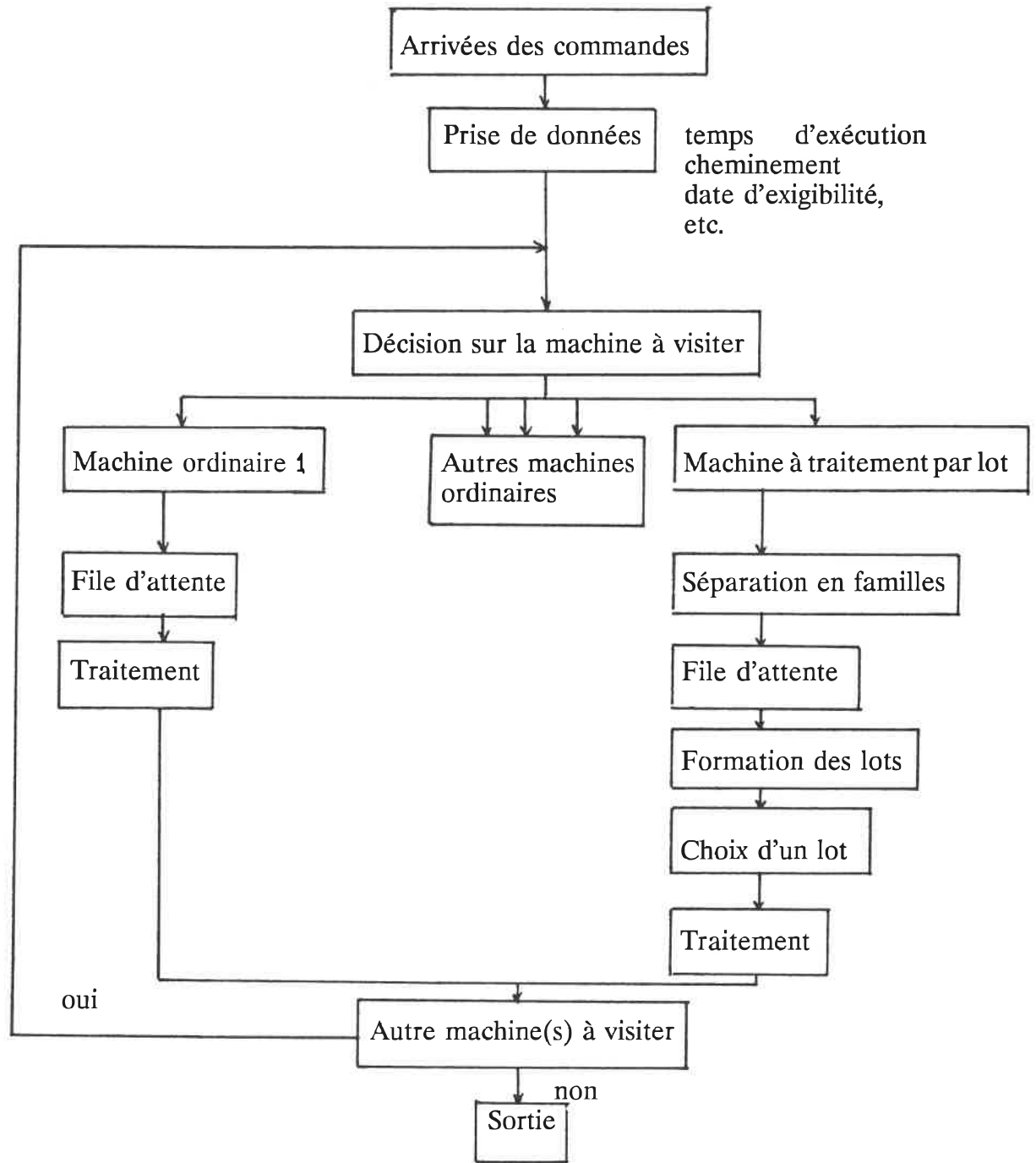


Fig. 6.1

Logique de programmation

6.3 ARRIVÉE DES COMMANDES

Comme pour les études antérieures, les temps inter-arrivée suivent une distribution exponentielle. On a fixé la moyenne à 1,1 parce que cela nous assurait un bon taux d'utilisation des machines de l'atelier ainsi qu'une stabilité des résultats. Les performances globales de l'atelier convergent assez rapidement vers une moyenne et n'ont pas de tendances ascendantes ou descendantes.

6.4 ASSIGNATION DES ATTRIBUTS

Chaque entité (pièce) qui circule dans le réseau transporte une série de valeurs qui lui sont propres. Ces valeurs sont appelées attributs dans le langage SLAM. La majorité de ces attributs sont assignés dès l'arrivée des pièces à l'atelier. Voici maintenant la description de ceux-ci:

- Les ATRIB(1) à ATRIB(10) désignent l'ordre par lequel les 10 machines seront visitées. Chacune des 10 machines n'est visitée qu'une seule fois.
- Les ATRIB(11) à ATRIB(19) représentent les temps d'exécution sur les 9 machines ordinaires. Les temps suivent une distribution exponentielle de moyenne 1.0 et chaque pièce créée aura ses propres temps.

- L'ATTRIB(20) représente le temps moyen de la machine à traitement par lot. On viendra changer la valeur de cet attribut pour rendre la machine à traitement par lot plus ou moins critique.
- L'ATTRIB(21) est simplement un compteur qui nous permet de savoir combien de machines la pièce a visitées.
- L'ATTRIB(22) nous aide à calculer le temps d'attente à la machine à traitement par lot. On assigne dans cet attribut le temps d'arrivée à la machine à traitement par lot.
- L'ATTRIB(23) représente le temps d'arrivée de la pièce à l'atelier.
- L'ATTRIB(24) représente la famille à laquelle la pièce fait partie. Nous avons 5 familles et chaque pièce à 20% de chance de faire partie de l'une d'elles.
- L'ATTRIB(25) représente le temps d'exécution de chacune des familles sur la machine à traitement par lot. Le temps varie en fonction du temps moyen fixé antérieurement (ATTRIB(20)).
- L'ATTRIB(26) est la date d'exigibilité de la pièce. Elle est fixée de façon proportionnelle à la somme des temps d'exécution.

- L'ATTRIB(27) est utilisé lorsque les lots de chaque famille pouvant être traités par la machine à traitement par lot sont formés. Il représente la valeur de celui-ci.
- Les ATTRIB(28) à ATTRIB(32) permettent de regrouper les pièces d'une même famille dans un lot.
- L'ATTRIB(33) permet de conserver tous les attributs de chacune des pièces qui ont formé le lot. Il est nécessaire car après le traitement, le lot sera séparé en pièces originales.
- L'ATTRIB(34) est utilisée pour 2 combinaisons de règles de priorité. Il sert soit de nombre aléatoire pour la combinaison RANDOM-RANDOM ou il représente la somme des temps d'exécution restant pour une pièce. Cette dernière valeur est utilisée pour calculer le SLACK.
- L'ATTRIB(35) représente pour chacune des pièces le temps d'attente à la machine à traitement par lot.
- L'ATTRIB(36) représente le SLACK. Cette valeur est calculé à l'arrivée d'une pièce à la machine à traitement par lot.

6.5 SOUS-PROGRAMMES

Plusieurs sous-programmes Fortran ont été utilisés pour améliorer l'efficacité de la programmation. Voici maintenant leur description sommaire:

- L'ÉVÉNEMENT 1 permet de déterminer le cheminement de chacune des pièces. Il est déterminé à l'arrivée et chaque pièce visitera chacune des machines une seule fois selon un ordre aléatoire sauf dans le cas où l'on placera la machine à traitement par lot en aval.
- L'ÉVÉNEMENT 2 permet de diriger une pièce vers la prochaine machine qui doit être visitée d'après son cheminement.
- L'ÉVÉNEMENT 3 sert dans la modélisation de la machine à traitement par lot. Étant donné que l'on forme un lot optimal pour chacune des familles et que l'on choisit par la suite qu'une seule famille, il est nécessaire après le traitement de la famille choisie, de remettre dans les files d'attente les pièces des familles non choisies.
- L'ÉVÉNEMENT 4 ne fait qu'ouvrir des GATES. Il est appelé après qu'un lot ait été traité, ce qui permet la formation de nouveaux lots. Dans le cas où l'on utilise la règle SLACK, cette valeur sera recalculée pour chacune des pièces étant donné qu'elle varie en fonction du temps.

- L'ÉVÉNEMENT 5 est utilisé lorsque l'on considère l'attente. S'il est préférable d'attendre on remettra en file d'attente les pièces des lots formés et on insérera un délai de 2 unités de temps avant de permettre la formation de nouveaux lots.
- L'ÉVÉNEMENT 6 est appelé après le délai prescrit dans l'événement 5. Il réinsère la machine à traitement par lot dans l'atelier ce qui permet d'appeler un autre sous-programme ALLOC(1) qui lui décide s'il est préférable d'attendre ou de traiter le lot immédiatement.
- L'ÉVÉNEMENT 7 ne fait que fermer les GATES, ce qui empêche la formation de nouveaux lots.
- USERF(1) représente le temps de traitement de la machine à traitement par lot. Étant donné qu'il varie en fonction des familles, on fait appel à cette fonction.
- ALLOC(1) permet de calculer la productivité en fonction du temps pour décider en faveur ou non du départ immédiat de la machine à traitement par lot.

Il est à noter que les 3 derniers ÉVÉNEMENTS et ALLOC(1) sont utilisés seulement lorsqu'on considère l'attente et non dans les autres cas.

6.6 FORMATION DES LOTS

Après qu'un lot ait été traité par la machine à traitement par lot, la formation de nouveaux lots optimaux pour chacune des familles débute. Les lots sont formés selon la règle de priorité utilisée, soit RANDOM, DUE DATE ou SLACK. La grosseur maximale de chacun des lots est de 10 (capacité de la machine). On conserve la grosseur de ces lots dans les variables XX(1), XX(2), XX(3), XX(4) et XX(5). Lorsque le lot optimum est formé on lui assigne une valeur globale qui dépend encore de la règle de priorité utilisée pour le choix de la famille soit RANDOM, WORKLOAD, SPT ou AVGDUE DATE. Chaque lot est ensuite dirigé vers la machine à traitement par lot qui est modélisé en RESOURCE ayant une capacité de 1. Les lots non choisis seront défaits et les pièces seront réinsérées en file d'attente pour permettre éventuellement la formation de nouveaux lots optimaux.

6.7 DURÉE DE SIMULATION

Nous désirons obtenir des performances à long-terme de l'atelier. Pour ce faire, nous avons introduit un nombre assez important de pièces dans l'atelier. En effet la simulation s'arrête lorsque la 10 000^{ième} pièce quitte l'atelier. Le temps total de simulation tourne autour de 11 000 unités de temps. Pour éliminer les résultats transitoires causés par le démarrage à vide de l'atelier, nous avons commencé à les

compiler qu'après 4 000 unités de temps. De plus, en faisant des rapports intermédiaires à chaque 1 000 unités de temps par la suite, on a constaté que les résultats étaient stables.

CHAPITRE 7

ANALYSE DES RÉSULTATS

7.1 INTRODUCTION

Dans ce chapitre, on analysera les diverses performances obtenues selon nos trois principales étapes élaborées lors du plan d'expérience. Premièrement, on analysera les résultats des combinaisons de règles qui nous permettent de choisir la famille à traiter et quelles pièces formeront le lot. Ensuite, on vérifiera si un changement de position de la machine à traitement par lot dans le cheminement a un effet sur les résultats obtenus lors de la première étape. Finalement, on analysera les avantages et inconvénients que peut nous apporter une politique d'attente pour la machine à traitement par lot.

7.2 ANALYSE DES COMBINAISONS DE RÈGLES

Nous avons proposé dans un chapitre antérieur 5 combinaisons de règles de priorité. Une d'entre elle, le RANDOM-RANDOM a pour objectif de servir principalement de base de comparaison pour les autres combinaisons. On pourra

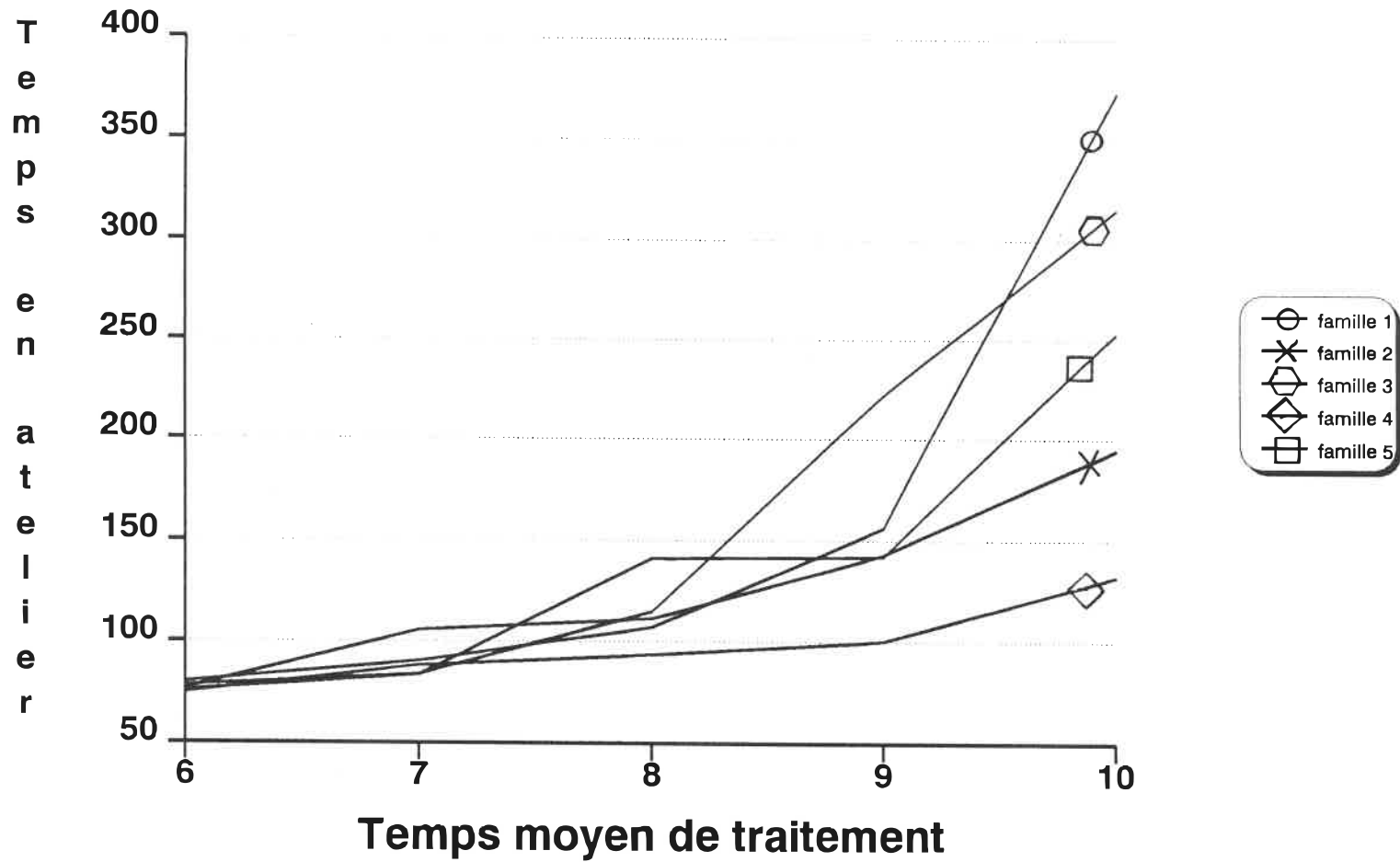
apprécier directement l'effet sur les performances globales d'utiliser les règles logiques car la combinaison RANDOM-RANDOM revient en fait à utiliser aucune méthode d'ordonnement et on peut facilement s'attendre à priori à de piètres résultats. Analysons maintenant les résultats de chacune des combinaisons.

7.2.1 RANDOM-RANDOM

Sur la figure 7.1 sont représentés les temps moyen en atelier (flow time) pour chacune des 5 familles en fonction du temps moyen de la machine à traitement par lot. On constate que plus le temps moyen de cette machine augmente plus il y a une divergence entre les temps en atelier des familles. Les positions des familles sont dues au hasard. L'augmentation de la variance du temps en atelier est due à l'augmentation du temps d'attente et de service sur la machine à traitement par lot qui sont très variables avec RANDOM-RANDOM par rapport au temps total en atelier qui comporte moins de variance. Une deuxième simulation utilisant un germe différent a donné des résultats passablement différents quant aux temps en atelier pour chaque famille. Cependant le temps moyen des 5 familles combinées est demeuré assez similaire. Nous avons aussi étudié un cas encore plus critique, i.e. fixer le temps moyen de la machine à traitement par lot à 11. Ce dernier cas a été trop critique et l'atelier n'était pas en mesure avec RANDOM-RANDOM d'avoir un taux de sortie des pièces équivalent au taux d'arrivée. L'atelier s'est donc congestionné assez rapidement.

Fig. 7.1

Temps en atelier avec RANDOM-RANDOM



Malgré les mauvais résultats de cette combinaison de règles, nous allons utiliser les performances moyennes obtenues pour comparer les autres combinaisons étudiées.

7.2.2 DUEDATE-SPT

Contrairement à ce que l'on avait espéré, cette combinaison s'est avérée totalement inefficace. La raison est pourtant fort simple. Notre machine à traitement par lot telle que nous l'avons modélisée est fortement utilisée. Il y a constamment des pièces en attente devant elle. La règle SPT a pour but de choisir la famille ayant le temps d'exécution le plus court. Les familles ayant des temps de traitement longs ne sont presque jamais choisies ou très rarement. Leur file d'attente se sont donc constamment allongées alors que les familles ayant de petits temps ont toujours eu de très faibles files d'attente.

Cette combinaison est donc inadéquate pour une machine à traitement par lot comme la nôtre même si dans les recherches antérieures, SPT a constamment offert des bons résultats par des machines dites ordinaires.

7.2.3 DUEDATE-AVGDUEDATE, DUEDATE-WORKLOAD ET SLACK-WORKLOAD

Les trois combinaisons ont offert des performances très supérieures par rapport à la combinaison RANDOM-RANDOM. Nous allons maintenant analyser leurs performances en fonction de plusieurs critères souvent utilisés en ordonnancement de production.

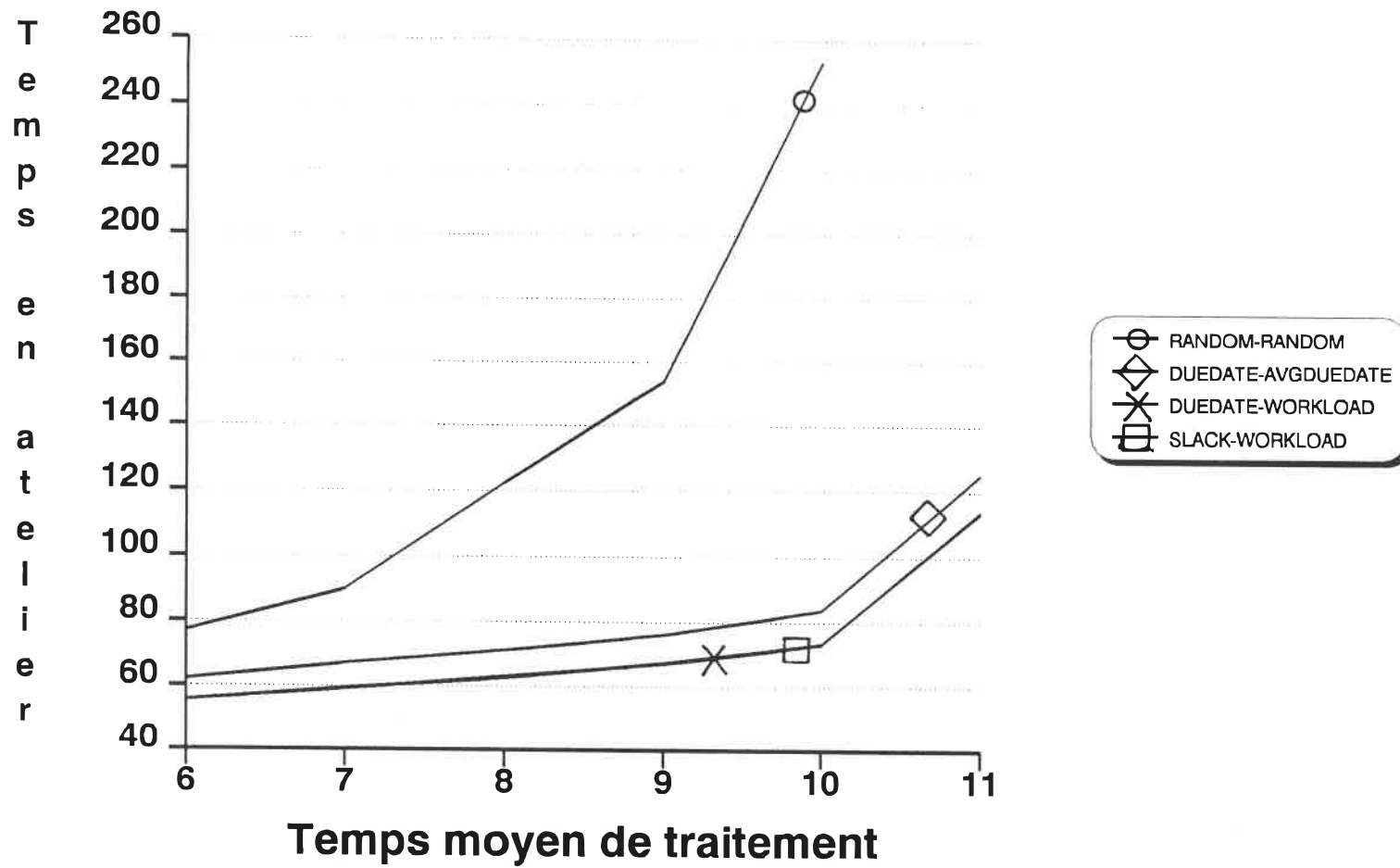
- Temps en atelier (flow time)

La figure 7.2 représente le temps moyen en atelier en fonction du temps moyen de la machine à traitement par lot. On constate que les combinaisons DUEDATE-WORKLOAD ET SLACK-WORKLOAD offrent des résultats presque identiques (les deux courbes sont superposées sur le graphique). La combinaison DUEDATE-AVGDUEDATE présente des résultats un peu moins bons. Les temps en atelier sont supérieurs d'environ 10%.

- Écart exigibilité terminaison (lateness)

Le respect des dates d'exigibilité est un critère important pour un atelier de sous-traitance car le non respect de celles-ci peut parfois occasionner des pénalités.

Fig 7.2
Temps en atelier



La figure 7.3 représente l'écart exigibilité-terminaison en fonction du temps moyen de la machine à traitement par lot. Nous retrouvons des résultats similaires à ceux obtenus pour le temps en atelier. Les combinaisons DUEDATE-WORKLOAD et SLACK-WORKLOAD offrent des performances très proches l'une de l'autre et la combinaison DUEDATE-AVGDUEDATE produit des écarts légèrement supérieurs.

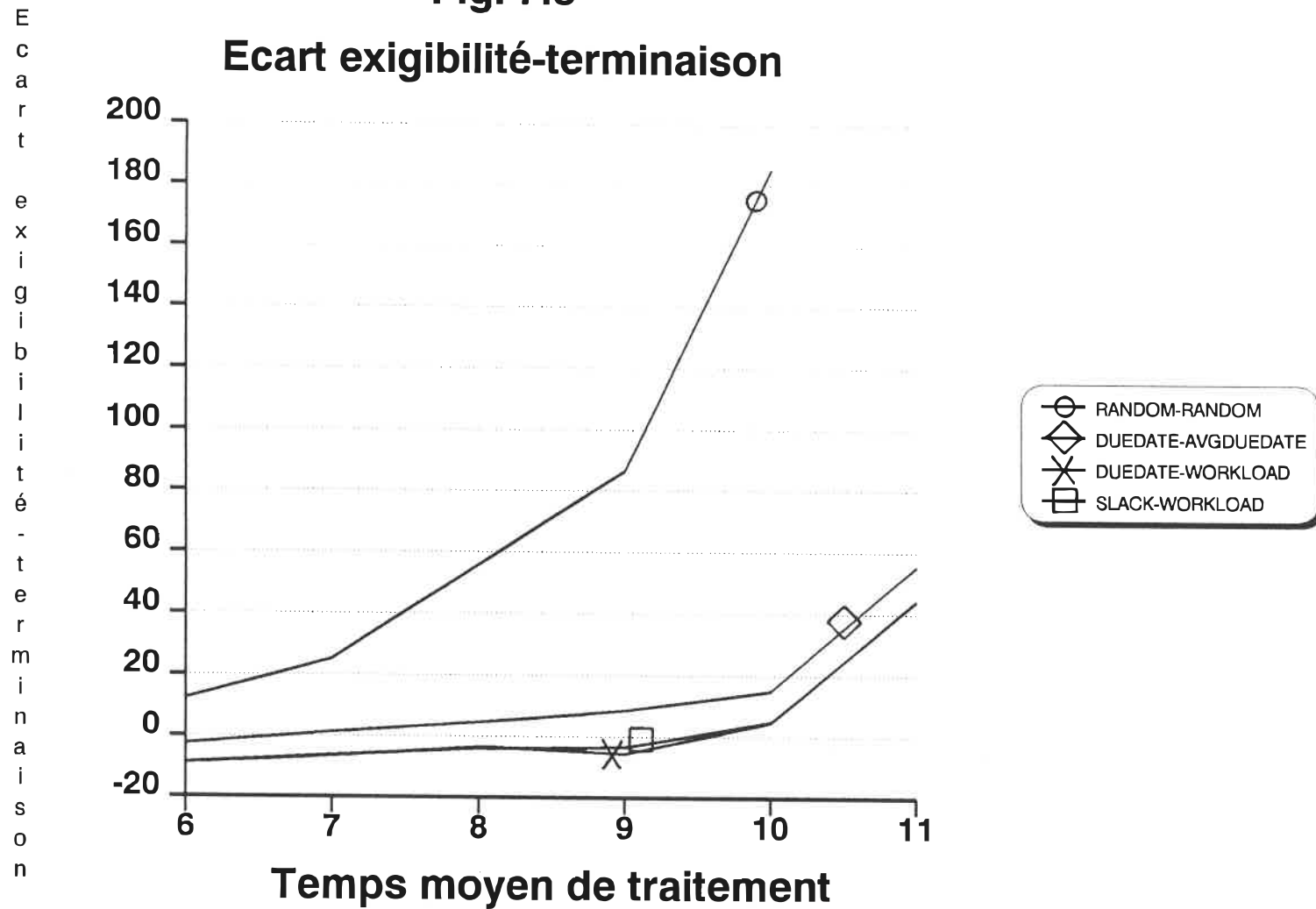
On peut expliquer les résultats des deux premières combinaisons mentionnées ci-dessus par le fait qu'elles visent à optimiser la productivité de la machine à traitement par lot en favorisant les familles ayant de longues files d'attente. Cela permet de libérer plus de pièces par la suite à l'atelier.

- Pourcentage de pièces en retard

Comme dit précédemment le respect des dates d'exigibilité est un critère important. Cependant l'écart exigibilité-terminaison n'indique pas tout. Il est bien d'avoir un écart faible (le plus négatif possible), mais on doit cependant vérifier le nombre de pièces qui sont terminées en retard par rapport à la date d'exigibilité prescrite.

Fig. 7.3

Ecart exigibilité-terminaison



La figure 7.4 représente le pourcentage de pièces en retard par rapport au temps moyen de la machine à traitement par lot. On constate ici que la combinaison SLACK-WORKLOAD performe mieux que DUEDATE-WORKLOAD lorsque le temps moyen de traitement est élevé. Le SLACK étant une sorte de date d'exigibilité révisée, certaines pièces peuvent devenir beaucoup plus critiques que d'autres. Le SLACK augmentera donc leur priorité et forcera ces pièces à être traitées en premier.

- Coût d'utilisation de la machine à traitement par lot

Ces trois premiers critères sont ceux le plus couramment utilisés dans les ateliers de sous-traitance. Notre cas est légèrement différent. La machine à traitement par lot peut avoir un coût de fonctionnement très élevé et il peut être non économique de traiter de petits lots. Un critère de performance supplémentaire à considérer dans ce cas est le nombre de lots traités. On constate à la figure 7.5 que le nombre de lots traités diminue à mesure que le temps moyen de traitement augmente. Cela s'explique que plus le temps est long, plus les files d'attente peuvent s'allonger. On a donc moins de lots mais ils sont par contre plus gros. On constate une plus grande différence en terme de lots traités lorsque le temps moyen de traitement est faible. Étant donné que la combinaison DUEDATE - AVGDUATE ne tient pas compte intrinsèquement de la grosseur des lots, on peut logiquement s'attendre à ce qu'elle nécessite plus de lots à traiter par rapport aux règles DUEDATE - WORKLOAD et SLACK - WORKLOAD qui choisiront les

Fig. 7.4

Pourcentage de pièces en retard

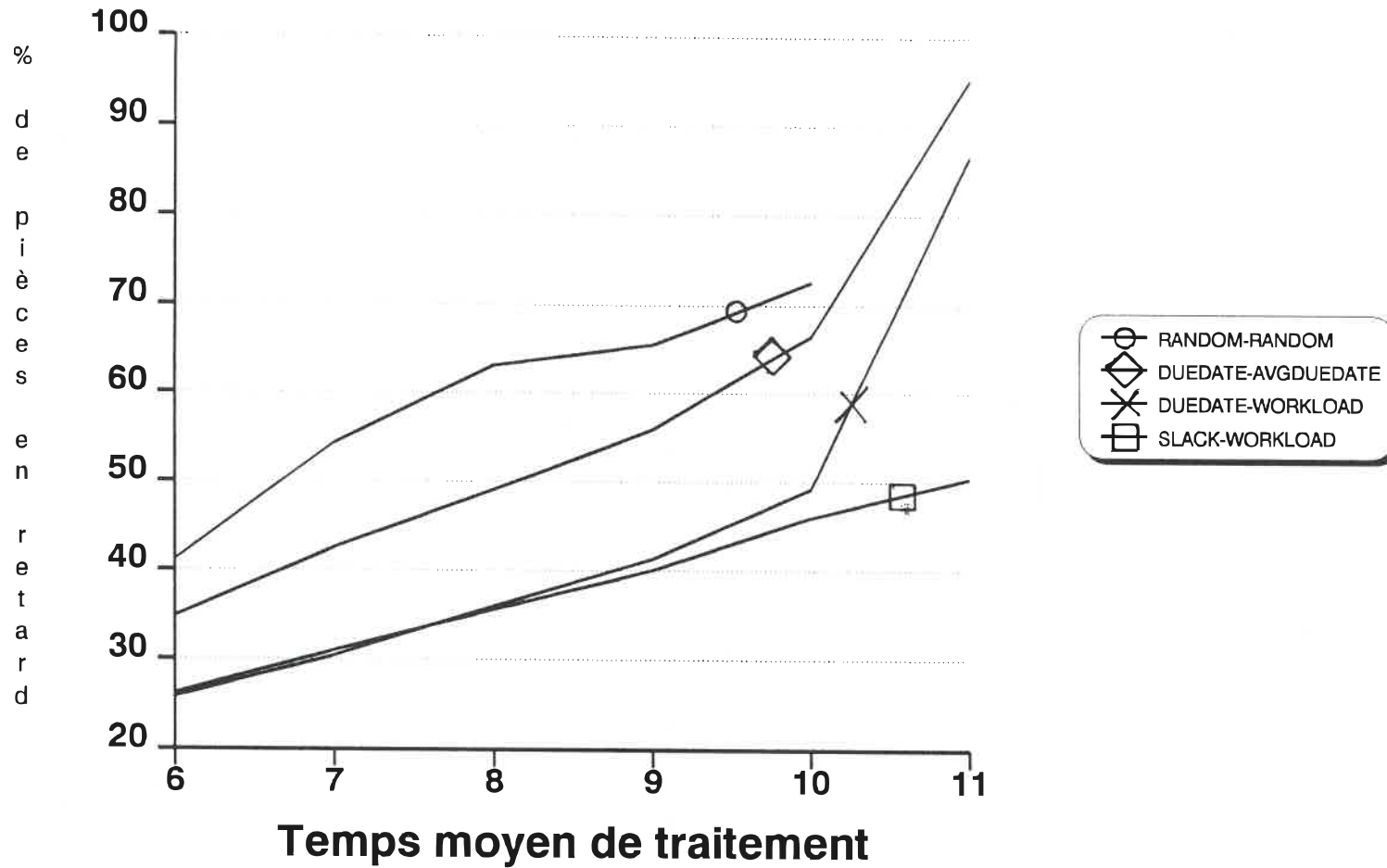
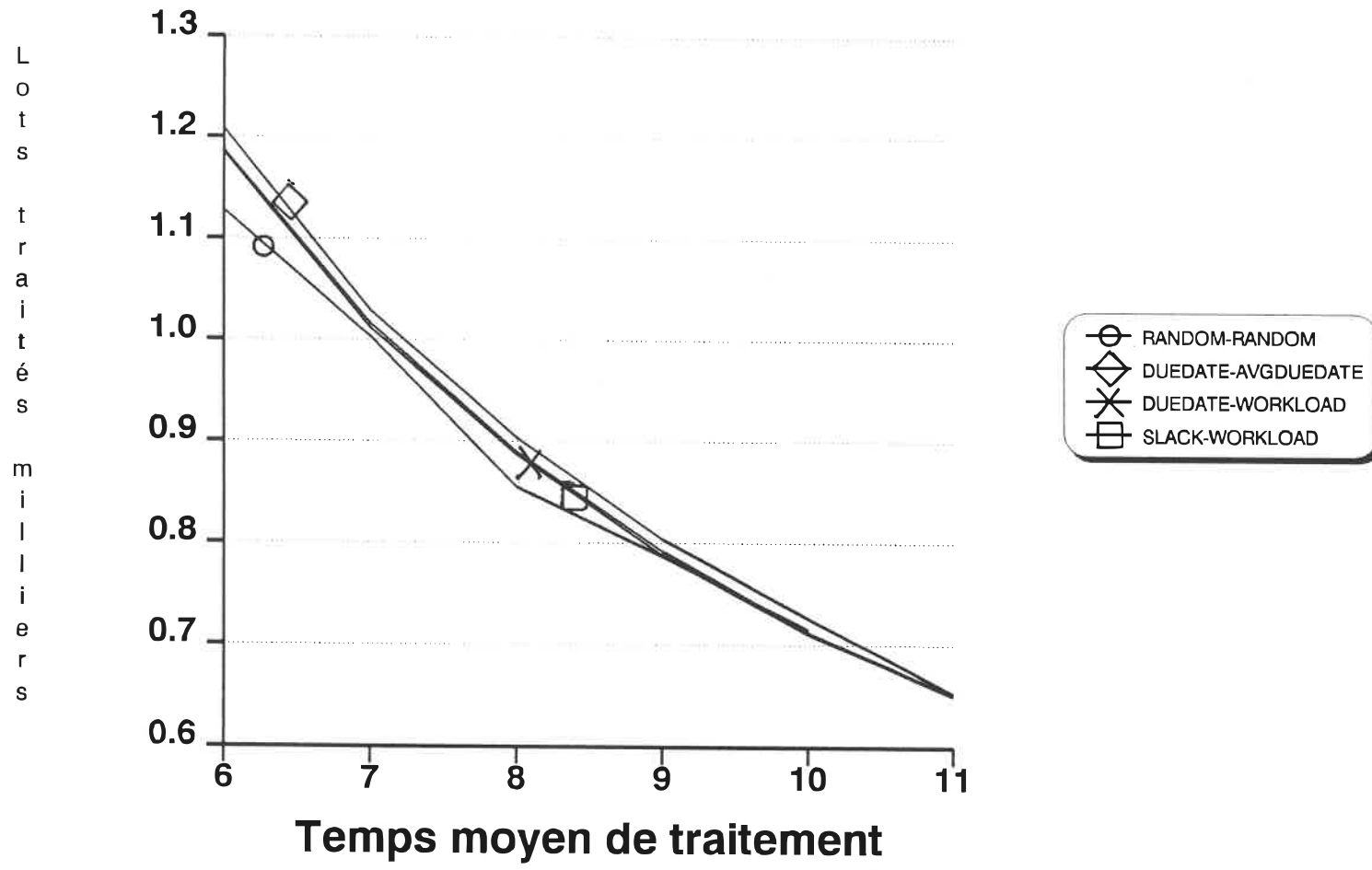


Fig. 7.5

Nombre de lots traités



files d'attente ayant le plus de pièces, ce qui permet éventuellement de traiter l'ensemble des pièces en un plus petit nombre de lots.

- Critères de performances secondaires

En plus d'analyser les performances globales de l'atelier, on peut regarder plus en détail s'il y a une divergence entre les performances des cinq familles. Le tableau 7.1 présente les résultats des diverses familles qui ont été obtenus en utilisant la combinaison SLACK-WORKLOAD et avec un temps moyen de 10 pour la machine à traitement par lot. On constate que les résultats sont similaires d'une famille à l'autre. Le temps en atelier varie quelque peu, cependant cela est logique étant donné que le temps de traitement sur la machine à traitement par lot varie d'une famille à l'autre. Finalement mentionnons que dans tous les cas analysés jusqu'à présent, la machine à traitement par lot était constamment en utilisation.

	Familles de pièces				
	1	2	3	4	5
Temps en atelier	75,2	72,1	72,8	78,7	68,9
Écart exigibilité-terminaison	4,0	5,6	4,7	5,4	4,3
Grosueur des lots	8,9	9,0	9,1	9,0	9,0
Attente moyenne à machine à traitement par lot	27,3	27,6	26,7	28,2	27,6
Nombre de lots traités	141	142	150	138	141
Nombre de pièces en retard	666	678	764	652	703
Nombre de pièces traitées	597	592	598	589	571

Tableau 7.1

Performances par familles de pièces selon DUE DATE-WORKLOAD

7.3 POSITION EN AVAL DE LA MACHINE À TRAITEMENT PAR LOT

On a voulu étudier si le positionnement de la machine à traitement par lot en aval avait un effet sur les performances des combinaisons. Pour ce faire nous avons pris les deux meilleures combinaisons de la section précédente soient, DUE DATE-WORKLOAD et SLACK-WORKLOAD et avons placé la dite machine

en 8^{ième} position du cheminement pour chaque pièce, les autres machines étant visitées aléatoirement. Nous avons considéré que 3 temps moyens différents; 9, 10 et 11. Tout cela ayant pour but de rendre la machine à traitement par lot la plus critique possible.

Le tableau 7.2 compare les résultats obtenus à la section précédente avec ceux obtenus en plaçant la machine en 8^{ième} place. On constate que les résultats sont très similaires, ce qui nous laisse croire qu'une position prédéterminée de la machine à traitement par lot n'a pas d'effet direct sur les performances de ces deux combinaisons.

7.4 POLITIQUE D'ATTENTE

Dans les cas où les coûts de fonctionnement de la machine à traitement par lot sont élevés, il peut être indésirable de la faire fonctionner à faible capacité. Il peut s'avérer préférable d'attendre pour permettre à des pièces d'arriver à cette machine pour former de plus gros lots même si certaines mesures de performance peuvent en souffrir, tel le temps en atelier, l'écart exigibilité-terminaison etc.

Dans les exécutions précédentes, on a pu constater que le taux de remplissage de la machine variait en fonction du temps moyen de traitement. Le tableau 7.3 résume les taux:

Combinaison		cas	Temps moyen de traitement		
			9	10	11
DUEDATE- WORKLOAD	Temps en atelier	1	67	73,2	114
		2	66,7	72,5	112,3
	Écart exigibilité terminaison	1	-5,5	4,6	44,4
		2	-0,9	3,9	42,7
	% de pièces en retard	1	41,4	49,2	86,6
		2	40,7	48,4	87,6
	Nombre de lots traités	1	788	711	651
		2	792	710	649
SLACK- WORKLOAD	Temps en atelier	1	67,3	73,5	113,6
		2	66,7	72,1	113,9
	Écart exigibilité terminaison	1	-34	4,9	44
		2	-0,9	3,5	44,3
	% de pièces en retard	1	40,1	46	50,4
		2	40,0	44,8	50,2
	nombre de lots traités	1	792	712	651
		2	789	711	651

- 1: Résultats obtenus antérieurement
2: Machine à traitement par lot fixée en 8^{ème} position

Tableau 7.2

Comparaison des performances avec machine à traitement par lot en 8^{ème} position

Temps de traitement moyen	Taux de remplissage
6	53 %
7	63 %
8	72 %
9	80 %
10	90 %
11	> 99 %

Tableau 7.3

Taux de remplissage en fonction du temps de traitement moyen

On analysera l'effet de l'attente pour les temps moyen 6, 7, 8, et 9 seulement étant donné que pour les 2 derniers temps, la machine est déjà fortement utilisée et qu'une attente n'améliorera pas les résultats.

Tel que mentionné au chapitre 4 (pages 41 et 42), on utilisera 2 méthodes différentes; une basée sur un coût fixe de traitement (Prod 1) et une autre basée sur le temps d'exécution de la famille en tant que telle (Prod 2). Les résultats de la première méthode sont présentés dans le tableau 7.4. Pour les besoins de la cause on a fixé le coût à 20. Quoique la fixation du coût soit libre à l'utilisateur, nous avons décidé de fixer le coût à 20. Ce coût nous assure que les lots formés seront toujours de grosseur raisonnable (minimum de 5 pièces). Un coût plus élevé créera des lots plus gros et un coût faible permettra des petits lots. Les résultats sont pour la combinaison DUE DATE-WORKLOAD. On constate avec cette méthode, que nous n'avons qu'une diminution sensible du nombre de lots traités et que cette diminution est très faible lorsque le temps moyen est de 8 ou 9. En somme les résultats sont un peu décevants.

	cas	Temps moyen de traitement			
		6	7	8	9
Temps en atelier	1	55,7	59,1	63,1	67
	2	57,3	61	63,6	68,2
Écart exigibilité-terminaison	1	-8,9	-6,5	-3,5	-5,5
	2	-7,4	-4,7	-2,9	-0,6
% de pièces en retard	1	25,9	30,4	36	41,4
	2	29,7	33,6	37,2	42,5
Nombre de lots traités	1	1185	1012	887	788
	2	1024	949	855	773
% d'utilisation de la machine	1	100	100	100	100
	2	86	92,6	95,8	97,3

1: sans attente

2: avec attente

Tableau 7.4

Comparaison des résultats de la combinaison DUEDATE-WORKLOAD en utilisant la formule Prod 1

Nous allons maintenant analyser la 2^{ème} méthode (Prod 2). Le paramètre α sera fixé à 20. Comme pour la situation précédente, un α de 20 nous assure d'avoir des lots de grosseur raisonnable. On a obtenu les résultats pour des combinaisons DUEDATE-WORKLOAD et SLACK-WORKLOAD. Ils sont résumés au tableau 7.5.

On constate une nette diminution du coût en terme de lots traités et de pourcentage d'utilisation de la machine. Par contre le temps en atelier sont moins bons et les dates d'exigibilité moins bien respectées. Il semble que lorsque le temps moyen de traitement est faible 6 ou 7, l'attente devient favorable étant donné que le nombre de lots traités diminue d'environ 25 % alors que les temps en atelier n'augmentent que de 10 % environ. On ne peut que faire cependant des analyses qualitatives lorsque l'on considère l'attente car nous ne connaissons pas le coût réel de fonctionnement de l'atelier. On sait cependant que la deuxième méthode (Prod 2) est plus efficace que la première. Une dernière caractéristique de cette méthode est que la grosseur moyenne des lots varie proportionnellement avec le temps de traitement. Cela à notre point de vue est un avantage.

Combinaison		cas	Temps moyen de traitement			
			6	7	8	9
DUEDATE- WORKLOAD	Temps en atelier	1	55,7	59,1	63,1	67
		2	63,7	66,9	71,6	74,4
	Écart exigibilité terminaison	1	-8,9	-6,5	-3,5	-5,5
		2	-8,5	1,3	5,0	6,8
	% de pièces en retard	1	25,9	30,4	36	41,4
		2	40,4	43,4	48,4	51,9
	Nombres de lots traités	1	1185	1012	887	788
		2	789	756	698	676
	% d'utilisation de la machine	1	100	100	100	100
		2	64,5	73,0	77,2	84,5
SLACK- WORKLOAD	Temps en atelier	1	55,8	59,4	62,7	67,3
		2	62,4	65,9	71,6	73,8
	Écart exigibilité terminaison	1	-8,8	-6,2	-3,4	-3,4
		2	-1,3	0,3	4,9	6,2
	% de pièces en retard	1	26,3	31	35,6	40,1
		2	39,7	42,1	47,2	47,2
	Nombre de lots traités	1	1187	1017	890	792
		2	798	752	700	679
	% d'utilisation de la machine	1	100	100	100	100
		2	65,4	72,6	77,2	84,7

- 1: sans attente
2: avec attente

Tableau 7.5

**Comparaison des résultats des combinaisons DUEDATE - WORKLOAD et
SLACK - WORKLOAD en considérant l'attente selon Prod 2**

7.5 RECOMMANDATIONS

Premièrement en ce qui a trait aux combinaisons de règles de priorité, deux d'entre elles se sont avérées supérieures: DUE DATE-WORKLOAD et SLACK-WORKLOAD. Cependant la deuxième a été supérieure à la première en ce qui a trait au pourcentage de pièces en retard lorsque le temps moyen de la machine à traitement par lot était long (9, 10, 11). Quoique le SLACK soit un peu plus difficile à calculer, il s'est avéré peu coûteux en terme de temps d'ordinateur. Notre recommandation est donc d'utiliser la combinaison SLACK-WORKLOAD peu importe le temps moyen de traitement de la machine à traitement par lot. Cette combinaison à l'avantage de vouloir maximiser la productivité de la machine à traitement par lot et en même temps tient compte d'une date d'exigibilité révisée, ce qui permet de mieux respecter les dates prescrites par les donneurs d'ordres.

On a pu aussi voir lors de la deuxième partie que le positionnement de la machine à traitement par lot en aval ne modifiait pas les résultats obtenus dans la première partie. Les 2 combinaisons sont donc assez stables et peuvent être utilisées semble-t-il peu importe où cette machine est positionnée.

En ce qui concerne l'attente, il nous est impossible de faire une évaluation quantitative réelle de ses coûts et bénéfices. On a par contre vu que la deuxième approche (Prod 2) peut réduire substantiellement le nombre total de lots traités. Dans le cas où le coût d'utilisation de cette machine est très élevé, cette approche

peut s'avérer justifiable. Nous croyons avoir atteint notre but en ce qui concerne l'attente en élaborant une méthode où l'utilisateur peut changer facilement un paramètre qui permet de représenter le type de machine qu'il a à sa disposition. La méthode proposée est donc assez générale.

CONCLUSION

Notre étude s'est voulue d'être une première approche au problème d'ordonnancement de la production sur commande d'un atelier comprenant une machine à traitement par lot. On a pu démontrer qu'un choix judicieux de règles de priorité pouvait améliorer les performances à long terme de l'atelier. Il s'est avéré qu'une combinaison considérant en même temps la productivité de la machine à traitement par lot ainsi qu'une certaine date d'exigibilité modifiée offrait d'excellents résultats. On a pu aussi démontrer qu'il était possible d'avoir comme objectif secondaire de diminuer le coût total de fonctionnement de cette machine en étant plus flexible en ce qui a trait au temps en atelier ainsi que par rapport aux dates d'exigibilité.

Nous avons restreint aussi le nombre de paramètres tels le nombre de familles de pièces, les temps de traitement, etc. Nous avons fait cela dans le but d'avoir un nombre raisonnable de situations à analyser et pour obtenir des résultats qu'on pourrait appeler de base.

Des études ultérieures pourraient analyser d'autres variantes de paramètres et même introduire certaines autres contraintes comme par exemple avoir des temps de mise en route dépendants de la séquence par laquelle les familles sont exécutées par la machine à traitement par lot.

Les décisions que nous avons prises pour ordonnancer la machine à traitement par lot sont d'ordre tactique. Il pourrait être intéressant d'étudier si une planification globale de production dans l'atelier de sous-traitance en vue d'optimiser la productivité de la machine à traitement par lot pourrait être bénéfique. Par exemple, on pourrait décider à priori d'un ordre sur lequel les familles seraient traitées. On pourrait aussi assigner différentes priorités aux pièces qui entrent dans l'atelier, ainsi que sur les machines en amont pour tenter de synchroniser la machine à traitement par lot. Cela demande cependant une approche différente du problème et est aussi beaucoup plus difficile.

BIBLIOGRAPHIE

- 1 - Centre National de la sous-traitance. Charte de la sous-traitance. 11, Avenue Hoche, 75 382 Paris, Cedex 08.
- 2 - Baker, Kenneth R. Introduction to Sequencing and Scheduling. John Wiley & Sons Inc. Toronto, 1974, 305 p.
- 3 - Conway, R.W., Maxwell, W.L., Miller, L.W. Theory of Scheduling. Addison-Wesley, Ontario, 1967, 294 p.
- 4 - Hax, Arnaldo, C., Candea Dan. Production and Inventory Management. Prentice-Hall inc, New Jersey 1984, 513 p.
- 5 - Blazewicz, J. Selected Topics in Scheduling Theory. Annals of Discrete Mathematics, 1987, p.1-60.
- 6 - French, S. Sequencing and Scheduling, Wiley, New York, 1982.
- 7 - Lawler, E.L. Recent Results in the Theory of Machine Scheduling, in A. Backem, M.Grotschel, B. Korte (eds.), Math. Programming: The State of the Art, Springer, 1982, p.202-234.

- 8 - Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. Recent Developments in Deterministic Sequencing: A Survey in: Deterministic and Stochastic Scheduling, M.A.H. Dempster et al. (eds.), Dordrecht, 1982, p.35-73.
- 9 - Lenstra, J.K. and Rinnooy Kan, A.H.G. Sequencing and Scheduling, in: M. OhEigearthaigh, J.K. Lenstra, A.H.G. Rinnooy Kan (eds.), Combinatorial Optimization: Annotated Bibliographies J. Wiley, Chichester, 1985, p.164-189.
- 10- Rinnooy Kan, A.H.G. Machine Scheduling Problems Classification Complexity and Computations, Nijhoff, The Hague, 1976.
- 11- Carlier, J. and Pinson, E. An Algorithm for solving the Job Shop Problem, Management Science, vol. 35, no. 2, 1989, p.164-176.
- 12- Pinedo, M. and Schrage, L. Stochastic Shop Scheduling: A Survey, in: Deterministic and Stochastic Scheduling, M.A.H. Dempster et al. (eds.), Dordrecht, 1982, p.181-196.
- 13- Blackstone, J.H., Phillips, D.T. and Hogg G.L. A state-of-the-art survey of dispatching rules for manufacturing job shop operations, International Journal of Production Research, vol. 20, no. 1, 1982, p.27-45.

- 14 - Panwalkar, S.S., Iskander, Wajik. A Survey of Scheduling Rules. Operations Research, vol. 25, no 1, January - February 1977. p.45-61.
- 15 - Nanot, Y.R. An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job Shop Like Queueing Networks Ph.D dissertation, UCLA, 1963.
- 16 - Maxwell, W.L., Mehra, M. Multiple-Factor Rules for Sequencing with Assembly Constraints. Naval Research Logistics Quarterly, Vol. 15 No 2, June 1968, p. 241-254.
- 17 - Neimeier, H.A., An Investigation of Alternative Routing in a Job Shop. Unpublished master's thesis, Cornell University, Ithaca, N.Y. June 1967.
- 18 - Wayson, R.D. The Effects of Alternative Machines on two Priority Dispatching Disciplines in the General Job Shop. Master's Thesis, Cornell University, Ithaca, N.Y., February 1965.
- 19 - Pritsker, Alan B. Simulation and SLAM II. Systems Publishing Corporation, West Lafayette, 1986. p. 839.

ANNEXE

LEVEL 2.3.0 (MAR 1988)

VS FORTRAN

SEP 06, 1988 11:06:25

REQUESTED OPTIONS (EXECUTE): LANGLVL(66)

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED TRMFLG SRCFLG NOSYM NORE
 SOUNP(TSN) NOSXM NOVECTOR IL(DIM) NOTEST NODC NOICA NODIRECTIVE NODRCS NOSAA
 OPT(C) LANGLVL(66) NOFIPS FLAG(I) AUTOBL(NONE) NAME(MAIN) LINECOUNT(60)

IF DD TSN *...*...1.....2.....3.....4.....5.....6.....7.*.....8

```

1      DIMENSION NSET(25000)
2      COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
3      1,NCRDR,NPRNT,NNRIN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
4      COMMON QSET(25000)
5      EQUIVALENCE(NSET(1),QSET(1))
6      NNSET=25000
7      NCRDR=5
8      NPRNT=6
9      NTAPE=7
10     CALL SLAM
11     STOP
11     END

```

STATISTICS SOURCE STATEMENTS = 11, PROGRAM SIZE = 1064 BYTES, PROGRAM NAME = MAIN PAGE: 1.

STATISTICS NO DIAGNOSTICS GENERATED.

MAIN** END OF COMPILATION 1 **

LEVEL 2.3.0 (MAR 1986)

VS FORTRAN

SEP 06, 1988 11:06:25

OPTIONS IN EFFECT: NDLIST NOMAP NOXREF NOGDSTMT NODECK SOURCE TERM OBJECT FIXED TRMFLG SRCFLG NOSYM NOREN
SDUMP(15) NOSXM NOVECTOR IL(DIM) NOTEST NODC NOICA NODIRECTIVE NODBCS NOSAA
OPT(0) LANGLVL(66) NDFIPS FLAG(1) AUTODBL(NONE) NAME(MAIN) LINECOUNT(60)

IF DD ISN *.....1.....2.....3.....4.....5.....6.....7.*.....8

```

1      SUBROUTINE EVENT(I)
2      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
3      1,MCRDR,MPRNT,MNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
4      DIMENSION M(10),N(10),A(45),NSET(1)
5      COMMON OSET(1)
6      EQUIVALENCE (NSET(1),OSET(1))
7      GO TO (1,30,300,400),I
1 10  I=1,10
1 11  M(I)=1
1 12  CONTINUE
1 13  T=1.1
1 14  K=1.5
1 15  DO 20 I=1,10
1 16  J=UNFRM(0.1,T,1)*10
1 17  N(I)=M(J)
1 18  M(J)=M(K)
1 19  M(K)=N(I)
1 20  T=T-0.1
1 21  K=K-1
1 22  ATRIB(I)=N(I)
1 23  CONTINUE
1 24  RETURN
1 25  IF(ATRIB(21).EQ.1.)GO TO 40
1 26  IF(ATRIB(21).EQ.2.)GO TO 60
1 27  IF(ATRIB(21).EQ.3.)GO TO 80
1 28  IF(ATRIB(21).EQ.4.)GO TO 100
1 29  IF(ATRIB(21).EQ.5.)GO TO 120
1 30  IF(ATRIB(21).EQ.6.)GO TO 140
1 31  IF(ATRIB(21).EQ.7.)GO TO 160
1 32  IF(ATRIB(21).EQ.8.)GO TO 180
1 33  IF(ATRIB(21).EQ.9.)GO TO 200
1 34  IF(ATRIB(21).EQ.10.)GO TO 220
1 35  IF(ATRIB(21).EQ.11.)GO TO 240
1 36  IF(ATRIB(1).EQ.1.)GO TO 245
1 37  IF(ATRIB(1).EQ.2.)GO TO 250
1 38  IF(ATRIB(1).EQ.3.)GO TO 255
1 39  IF(ATRIB(1).EQ.4.)GO TO 260
1 40  IF(ATRIB(1).EQ.5.)GO TO 265
1 41  IF(ATRIB(1).EQ.6.)GO TO 270
1 42  IF(ATRIB(1).EQ.7.)GO TO 275
1 43  IF(ATRIB(1).EQ.8.)GO TO 280
1 44  IF(ATRIB(1).EQ.9.)GO TO 285
1 45  IF(ATRIB(1).EQ.10.)GO TO 290
1 46  IF(ATRIB(2).EQ.1.)GO TO 245
1 47  IF(ATRIB(2).EQ.2.)GO TO 250
1 48  IF(ATRIB(2).EQ.3.)GO TO 255
1 49  IF(ATRIB(2).EQ.4.)GO TO 260
1 50  IF(ATRIB(2).EQ.5.)GO TO 265
1 51  IF(ATRIB(2).EQ.6.)GO TO 270
1 52  IF(ATRIB(2).EQ.7.)GO TO 275
1 53  IF(ATRIB(2).EQ.8.)GO TO 280
1 54  IF(ATRIB(2).EQ.9.)GO TO 285
1 55  IF(ATRIB(2).EQ.10.)GO TO 290

```

LEVEL 2.3.0 (MAR 1988)

VS FORTRAN

SEP 06, 1988 11:06:25 NAME:EVENT

IF DD	ISN	*.....*...1.....2.....3.....4.....5.....6.....7.*.....8
	52	IF (ATTRIB(2).EQ.10.)GO TO 290
	53	IF (ATTRIB(3).EQ.1.)GO TO 245
	54	IF (ATTRIB(3).EQ.2.)GO TO 250
	55	IF (ATTRIB(3).EQ.3.)GO TO 255
	56	IF (ATTRIB(3).EQ.4.)GO TO 260
	57	IF (ATTRIB(3).EQ.5.)GO TO 265
	58	IF (ATTRIB(3).EQ.6.)GO TO 270
	59	IF (ATTRIB(3).EQ.7.)GO TO 275
	60	IF (ATTRIB(3).EQ.8.)GO TO 290
	61	IF (ATTRIB(3).EQ.9.)GO TO 285
	62	IF (ATTRIB(3).EQ.10.)GO TO 290
	63	IF (ATTRIB(4).EQ.1.)GO TO 245
	64	IF (ATTRIB(4).EQ.2.)GO TO 250
	65	IF (ATTRIB(4).EQ.3.)GO TO 255
	66	IF (ATTRIB(4).EQ.4.)GO TO 260
	67	IF (ATTRIB(4).EQ.5.)GO TO 265
	68	IF (ATTRIB(4).EQ.6.)GO TO 270
	69	IF (ATTRIB(4).EQ.7.)GO TO 275
	70	IF (ATTRIB(4).EQ.8.)GO TO 280
	71	IF (ATTRIB(4).EQ.9.)GO TO 285
	72	IF (ATTRIB(4).EQ.10.)GO TO 290
	73	IF (ATTRIB(5).EQ.1.)GO TO 245
	74	IF (ATTRIB(5).EQ.2.)GO TO 250
	75	IF (ATTRIB(5).EQ.3.)GO TO 255
	76	IF (ATTRIB(5).EQ.4.)GO TO 260
	77	IF (ATTRIB(5).EQ.5.)GO TO 265
	78	IF (ATTRIB(5).EQ.6.)GO TO 270
	79	IF (ATTRIB(5).EQ.7.)GO TO 275
	80	IF (ATTRIB(5).EQ.8.)GO TO 280
	81	IF (ATTRIB(5).EQ.9.)GO TO 285
	82	IF (ATTRIB(5).EQ.10.)GO TO 290
	83	IF (ATTRIB(6).EQ.1.)GO TO 245
	84	IF (ATTRIB(6).EQ.2.)GO TO 250
	85	IF (ATTRIB(6).EQ.3.)GO TO 255
	86	IF (ATTRIB(6).EQ.4.)GO TO 260
	87	IF (ATTRIB(6).EQ.5.)GO TO 265
	88	IF (ATTRIB(6).EQ.6.)GO TO 270
	89	IF (ATTRIB(6).EQ.7.)GO TO 275
	90	IF (ATTRIB(6).EQ.8.)GO TO 280
	91	IF (ATTRIB(6).EQ.9.)GO TO 285
	92	IF (ATTRIB(6).EQ.10.)GO TO 290
	93	IF (ATTRIB(7).EQ.1.)GO TO 245
	94	IF (ATTRIB(7).EQ.2.)GO TO 250
	95	IF (ATTRIB(7).EQ.3.)GO TO 255
	96	IF (ATTRIB(7).EQ.4.)GO TO 260
	97	IF (ATTRIB(7).EQ.5.)GO TO 265
	98	IF (ATTRIB(7).EQ.6.)GO TO 270
	99	IF (ATTRIB(7).EQ.7.)GO TO 275
	100	IF (ATTRIB(7).EQ.8.)GO TO 280
	101	IF (ATTRIB(7).EQ.9.)GO TO 285
	102	IF (ATTRIB(7).EQ.10.)GO TO 290
	103	IF (ATTRIB(8).EQ.1.)GO TO 245
	104	IF (ATTRIB(8).EQ.2.)GO TO 250
	105	IF (ATTRIB(8).EQ.3.)GO TO 255
	106	IF (ATTRIB(8).EQ.4.)GO TO 260
	107	IF (ATTRIB(8).EQ.5.)GO TO 265

LEVEL 2.3.0 (MAR 1988)

VS FORTRAN

SEP 06, 1988 11:06:25 NAME:EVENT

IF DO ISN *.....1.....2.....3.....4.....5.....6.....7.*.....8

108		IF (ATTRIB(8).EQ.6.)GO TO 270
109		IF (ATTRIB(8).EQ.7.)GO TO 275
110		IF (ATTRIB(8).EQ.8.)GO TO 280
111		IF (ATTRIB(8).EQ.9.)GO TO 285
112	200	IF (ATTRIB(8).EQ.10.)GO TO 290
113		IF (ATTRIB(9).EQ.1.)GO TO 245
114		IF (ATTRIB(9).EQ.2.)GO TO 250
115		IF (ATTRIB(9).EQ.3.)GO TO 255
116		IF (ATTRIB(9).EQ.4.)GO TO 260
117		IF (ATTRIB(9).EQ.5.)GO TO 265
118		IF (ATTRIB(9).EQ.6.)GO TO 270
119		IF (ATTRIB(9).EQ.7.)GO TO 275
120		IF (ATTRIB(9).EQ.8.)GO TO 280
121		IF (ATTRIB(9).EQ.9.)GO TO 285
122		IF (ATTRIB(9).EQ.10.)GO TO 290
123	220	IF (ATTRIB(10).EQ.1.)GO TO 245
124		IF (ATTRIB(10).EQ.2.)GO TO 250
125		IF (ATTRIB(10).EQ.3.)GO TO 255
126		IF (ATTRIB(10).EQ.4.)GO TO 260
127		IF (ATTRIB(10).EQ.5.)GO TO 265
128		IF (ATTRIB(10).EQ.6.)GO TO 270
129		IF (ATTRIB(10).EQ.7.)GO TO 275
130		IF (ATTRIB(10).EQ.8.)GO TO 280
131		IF (ATTRIB(10).EQ.9.)GO TO 285
132		IF (ATTRIB(10).EQ.10.)GO TO 290
133	240	II=11
134		RETURN
135	245	II=1
136		RETURN
137	250	II=2
138		RETURN
139	255	II=3
140		RETURN
141	260	II=4
142		RETURN
143	265	II=5
144		RETURN
145	270	II=6
146		RETURN
147	275	II=7
148		RETURN
149	280	II=8
150		RETURN
151	285	II=9
152		RETURN
153	290	II=10
154		RETURN
155	300	IF (NMQ(20).EQ.0)GO TO 350
156		IF (NMQ(20).NE.1)GO TO 310
157		CALL REMOVE(1,20,A)
158		CALL ENTER(1,A)
159		GO TO 350
160	310	IF (NMQ(20).NE.2)GO TO 320
161		CALL REMOVE(1,20,A)
162		CALL ENTER(1,A)
163		CALL REMOVE(1,20,A)

LEVEL 2.2.0 (MAR 1988)

VS FORTRAN

SEP 06, 1988 11:06:25 NAME:EVENT

IF DO ISN *.....1.....2.....3.....4.....5.....6.....7.*.....8

```
164 CALL ENTER(1,A)
165 GO TO 350
166 320 IF(NNO(20).NE.3)GO TO 330
167 CALL REMOVE(1,20,A)
168 CALL ENTER(1,A)
169 CALL REMOVE(1,20,A)
170 CALL ENTER(1,A)
171 CALL REMOVE(1,20,A)
172 CALL ENTER(1,A)
173 GO TO 350
174 330 DO 340 I=1,4
175 CALL REMOVE(1,20,A)
176 CALL ENTER(1,A)
177 340 CONTINUE
178 350 XX(1)=0
179 XX(2)=0
180 XX(3)=0
181 XX(4)=0
182 XX(5)=0
183 RETURN
184 400 DO 410 I=10,14
185 NTRY=MMLE(I)
186 405 IF(NTRY.EQ.0)GO TO 410
187 NEXT=NPRED(NTRY)
188 QSET(NTRY+36)=(QSET(NTRY+26)-QSET(NTRY+34)-TNOW)/QSET(NTRY+34)
189 IF(QSET(NTRY+36).GE.0)GO TO 407
190 QSET(NTRY+36)=QSET(NTRY+34)/(QSET(NTRY+26)-QSET(NTRY+34)-TNOW)
191 407 CALL ULINK(-NTRY,I)
192 CALL LINK(I)
193 NTRY=NEXT
194 GO TO 405
195 410 CONTINUE
196 CALL OPEN(1)
197 CALL OPEN(2)
198 CALL OPEN(3)
199 CALL OPEN(4)
200 CALL OPEN(5)
201 RETURN
202 END
```

STATISTICS SOURCE STATEMENTS = 202, PROGRAM SIZE = 6876 BYTES, PROGRAM NAME = EVENT PAGE: 2.

STATISTICS NO DIAGNOSTICS GENERATED.

EVENT END OF COMPILATION 2 *****

LEVEL 2.3.0 (MAR 1983)

VS FORTRAN

SEP 06, 1988 11:06:25

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED TRMFLG SRCFLG NOSYM NORD
SDUMP(LSN) NOSYM NOVECTOR IL(DIM) NOTEST NOOC NOICA MODIRECTIVE NOOBCS NOSAA
OPT(0) LANGLVL(66) NOFIPS FLAG(I) AUTOOBL(NONE) NAME(MAIN) LINECOUNT(60)

IF DD ISN *.....1.....2.....3.....4.....5.....6.....7.*.....8

```

1 FUNCTION USERF(IFN)
2 COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
3 1,NCDR,NPRNT,NMRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
4 GO TO (1),IFN
5 CALL CLOSX(1)
6 CALL CLOSX(2)
7 CALL CLOSX(3)
8 CALL CLOSX(4)
9 CALL CLOSX(5)
10 USERF=ATTRIB(25)
11 RETURN
END

```

STATISTICS SOURCE STATEMENTS = 11, PROGRAM SIZE = 1128 BYTES, PROGRAM NAME = USERF PAGE: 6.

STATISTICS NO DIAGNOSTICS GENERATED.

USERF END OF COMPILATION 3 *****

```

1 GEN,RACINE,ATELIER,07/25/89,1:
2 LIM,20,36,500:
3 INILC,XX(1)=0,XX(2)=0,XX(3)=0,XX(4)=0,XX(5)=0:
4 PRIORITY/1,LVF(11)/2,LVF(12)/3,LVF(13)/4,LVF(14)/5,LVF(15)/6,LVF(16)/
5 7,LVF(17)/8,LVF(18)/9,LVF(19)/10,LVF(20)/11,LVF(21)/12,LVF(22)/13,
6 PRIORITY/17,LVF(36)/18,LVF(36)/19,LVF(36)/20,LVF(27)/10,LVF(36)/
7 11,LVF(36)/12,LVF(36)/13,LVF(36)/14,LVF(36):
8 PRIORITY/NCLNR,LVF(25):
9 NETWORK:
10 RESOURCE,MACH8(1),20:
11 GATES/LOT1,OPEN,10/LOT2,OPEN,11/LOT3,OPEN,12/LOT4,OPEN,13/
12 LOTS,OPEN,14:
13 CREATE,EXPON(1.1),.23:
14 EVENT,1:
15 ASSIGN,TRIB(11)=EXPON(1),
16 TRIB(12)=EXPON(1),
17 TRIB(13)=EXPON(1),
18 TRIB(14)=EXPON(1),
19 TRIB(15)=EXPON(1),
20 TRIB(16)=EXPON(1),
21 TRIB(17)=EXPON(1),
22 TRIB(18)=EXPON(1),
23 TRIB(19)=EXPON(1),
24 TRIB(20)=9:
25 ASSIGN,TRIB(21)=1:
26 GOOD,1:
27 ACT/1,..2,UN;FAM1
28 ACT/2,..2,DEUX;FAM2
29 ACT/3,..2,TROIS;FAM3
30 ACT/4,..2,QUAT;FAM4
31 ACT/5,..2,CINQ;FAM5
32 UN ASSIGN,TRIB(24)=1,
33 TRIB(25)=TRIB(20)*1.25;
34 ACT,..,DATE:
35 DEUX ASSIGN,TRIB(24)=2,
36 TRIB(25)=TRIB(20)*0.75;
37 ACT,..,DATE:
38 TROIS ASSIGN,TRIB(24)=3,
39 TRIB(25)=TRIB(20)*1.00;
40 ACT,..,DATE:
41 QUAT ASSIGN,TRIB(24)=4,
42 TRIB(25)=TRIB(20)*1.50;
43 ACT,..,DATE:
44 CINQ ASSIGN,TRIB(24)=5,
45 TRIB(25)=TRIB(20)*0.50;
46 DATE ASSIGN,ATR(26)=ATR(11)+ATR(12)+ATR(13)+ATR(14)+ATR(15)+ATR(16),
47 ATR(26)=ATR(26)+ATR(17)+ATR(18)+ATR(19),
48 ATR(26)=ATR(26)+ATR(25),
49 AAA EVENT,2;
50 GOOD,1:
51 ACT,..,II.EQ.1,M1:
52 ACT,..,II.EQ.2,M2:
53 ACT,..,II.EQ.3,M3:
54 ACT,..,II.EQ.4,M4:
55 ACT,..,II.EQ.5,M5:
56 ACT,..,II.EQ.6,M6:
57 ACT,..,II.EQ.7,M7:
58 ACT,..,II.EQ.8,M8:
59 ACT,..,II.EQ.9,M9:
60 ACT,..,II.EQ.10,M10:
61 ACT,..,II.EQ.11,OUT:
62 :
63 :
64 #1 QUEUE(1):

```



```

65      ACT/6, ATRIB(11); MACH1
66      ASSIGN, ATRIB(21)=ATRIB(21)+1,
67          ATRIB(34)=ATRIB(34)-ATRIB(11);
68      ACT,,,AAA;
69      ;
70      M2      QUEUE(2);
71      ACT/7, ATRIB(12); MACH2
72      ASSIGN, ATRIB(21)=ATRIB(21)+1,
73          ATRIB(34)=ATRIB(34)-ATRIB(12);
74      ACT,,,AAA;
75      ;
76      M3      QUEUE(3);
77      ACT/8, ATRIB(13); MACH3
78      ASSIGN, ATRIB(21)=ATRIB(21)+1,
79          ATRIB(34)=ATRIB(34)-ATRIB(13);
80      ACT,,,AAA;
81      ;
82      M4      QUEUE(4);
83      ACT/9, ATRIB(14); MACH4
84      ASSIGN, ATRIB(21)=ATRIB(21)+1,
85          ATRIB(34)=ATRIB(34)-ATRIB(14);
86      ACT,,,AAA;
87      ;
88      M5      QUEUE(5);
89      ACT/10, ATRIB(15); MACH5
90      ASSIGN, ATRIB(21)=ATRIB(21)+1,
91          ATRIB(34)=ATRIB(34)-ATRIB(15);
92      ACT,,,AAA;
93      ;
94      M6      QUEUE(6);
95      ACT/11, ATRIB(16); MACH6
96      ASSIGN, ATRIB(21)=ATRIB(21)+1,
97          ATRIB(34)=ATRIB(34)-ATRIB(16);
98      ACT,,,AAA;
99      ;
100     M7      QUEUE(7);
101     ACT/12, ATRIB(17); MACH7
102     ASSIGN, ATRIB(21)=ATRIB(21)+1,
103         ATRIB(34)=ATRIB(34)-ATRIB(17);
104     ACT,,,AAA;
105     ;
106     M8      QUEUE(8);
107     ACT/13, ATRIB(18); MACH8
108     ASSIGN, ATRIB(21)=ATRIB(21)+1,
109         ATRIB(34)=ATRIB(34)-ATRIB(18);
110     ACT,,,AAA;
111     ;
112     M9      QUEUE(9);
113     ACT/14, ATRIB(19); MACH9
114     ASSIGN, ATRIB(21)=ATRIB(21)+1,
115         ATRIB(34)=ATRIB(34)-ATRIB(19);
116     ACT,,,AAA;
117     ;
118     M10     ASSIGN, ATRIB(22)=TNOW;
119     ACT,,,M3;
120     XX      ENTER, 1;
121     YY      UNBATCH, 33;
122     MB      GOON, 1;
123     ACT/15, , ATRIB(24).EQ.1, AW1:FA1
124     ACT/16, , ATRIB(24).EQ.2, AW2:FA2
125     ACT/17, , ATRIB(24).EQ.3, AW3:FA3
126     ACT/18, , ATRIB(24).EQ.4, AW4:FA4
127     ACT/19, , ATRIB(24).EQ.5, AW5:FA5
128     ;
129     AW1     AWAIT(10), LOT1;
130     QUEUE(15), , 9, BALK(L1);
131     ACT/21, , 201, , CL1;
132     L1      CLOSE, LOT1;

```

```

133      ACT,,,AW1;
134  CL1  CLOSE,LOT1;
135      ASSIGN,XX(1)=XX(1)+1,1;
136      ACT,,,XX(1).LT.10.AND.NNO(15).GT.0.OR.NNACT(21).EQ.1,BBB;
137      ACT;
138      ASSIGN,TRIB(28)=-1;
139      ACT,,,B1;
140  BBB  ASSIGN,TRIB(28)=1;
141  B1   BATCH,1/28,,,FIRST,ALL(33);
142      ASSIGN,TRIB(27)=XX(1)+NNO(10);
143      ACT/31,,,FILE:GROUP1
144      ;
145  AW2  AWAIT(11),LOT2;
146      QUEUE(16),,9,BALK(L2);
147      ACT/22,0,001,,CL2;
148  L2  CLOSE,LOT2;
149      ACT,,,AW2;
150  CL2  CLOSE,LOT2;
151      ASSIGN,XX(2)=XX(2)+1,1;
152      ACT,,,XX(2).LT.10.AND.NNO(16).GT.0.OR.NNACT(22).EQ.1,CCC;
153      ACT;
154      ASSIGN,TRIB(29)=-1;
155      ACT,,,B2;
156  CCC  ASSIGN,TRIB(29)=1;
157  B2   BATCH,1/29,,,FIRST,ALL(33);
158      ASSIGN,TRIB(27)=XX(2)+NNO(11);
159      ACT/32,,,FILE:GROUP2
160      ;
161  AW3  AWAIT(12),LOT3;
162      QUEUE(17),,9,BALK(L3);
163      ACT/23,0,001,,CL3;
164  L3  CLOSE,LOT3;
165      ACT,,,AW3;
166  CL3  CLOSE,LOT3;
167      ASSIGN,XX(3)=XX(3)+1,1;
168      ACT,,,XX(3).LT.10.AND.NNO(17).GT.0.OR.NNACT(23).EQ.1,DDD;
169      ACT;
170      ASSIGN,TRIB(30)=-1;
171      ACT,,,B3;
172  DDD  ASSIGN,TRIB(30)=1;
173  B3   BATCH,1/30,,,FIRST,ALL(33);
174      ASSIGN,TRIB(27)=XX(3)+NNO(12);
175      ACT/33,,,FILE:GROUP3
176      ;
177  AW4  AWAIT(13),LOT4;
178      QUEUE(18),,9,BALK(L4);
179      ACT/24,0,001,,CL4;
180  L4  CLOSE,LOT4;
181      ACT,,,AW4;
182  CL4  CLOSE,LOT4;
183      ASSIGN,XX(4)=XX(4)+1,1;
184      ACT,,,XX(4).LT.10.AND.NNO(18).GT.0.OR.NNACT(24).EQ.1,EEE;
185      ACT;
186      ASSIGN,TRIB(31)=-1;
187      ACT,,,B4;
188  EEE  ASSIGN,TRIB(31)=1;
189  B4   BATCH,1/31,,,FIRST,ALL(33);
190      ASSIGN,TRIB(27)=XX(4)+NNO(13);
191      ACT/34,,,FILE:GROUP4
192      ;
193  AW5  AWAIT(14),LOTS;
194      QUEUE(19),,9,BALK(L5);
195      ACT/25,0,001,,CL5;
196  L5  CLOSE,LOTS;
197      ACT,,,AW5;
198  CL5  CLOSE,LOTS;
199      ASSIGN,XX(5)=XX(5)+1,1;
200      ACT,,,XX(5).LT.10.AND.NNO(19).GT.0.OR.NNACT(25).EQ.1,FFF;
201      ACT;

```

```

201      ACT,;
202      ASSIGN, ATRIB(32)=-1;
203      ACT,;,R5;
204 FFF      ASSIGN, ATRIB(32)=1;
205 B5      BATCH,1/32,;,FIRST,ALL(33);
206      ASSIGN, ATRIB(27)=XX(5)+NNQ(14);
207      ACT/35,;,FILE;GROUP5
208 FILE    AWAIT(20),MACHB/1;
209      ACT/20,USERF(1);MACHBAT
210      GOON,1;
211      ACT/36,;,ATRIB(24).EQ.1,P1;LOTFAM1
212      ACT/37,;,ATRIB(24).EQ.2,P2;LOTFAM2
213      ACT/38,;,ATRIB(24).EQ.3,P3;LOTFAM3
214      ACT/39,;,ATRIB(24).EQ.4,P4;LOTFAM4
215      ACT/40,;,ATRIB(24).EQ.5,P5;LOTFAM5
216 P1      COLCT,XX(1),BATCH1;
217      ACT,;,SS;
218 P2      COLCT,XX(2),BATCH2;
219      ACT,;,SS;
220 P3      COLCT,XX(3),BATCH3;
221      ACT,;,SS;
222 P4      COLCT,XX(4),BATCH4;
223      ACT,;,SS;
224 P5      COLCT,XX(5),BATCH5;
225 SS      EVENT,3;
226      ACT,0.02;
227      EVENT,4;
228      ACT,0.02;
229      FREE,MACHB/1;
230 ZZ      UNBATCH,33;
231      ASSIGN, ATRIB(21)=ATRIB(21)+1;
232      ATRIB(34)=ATRIB(34)-ATRIB(25);
233      GOON,1;
234      ACT,;,ATRIB(24).EQ.1,R1;
235      ACT,;,ATRIB(24).EQ.2,R2;
236      ACT,;,ATRIB(24).EQ.3,R3;
237      ACT,;,ATRIB(24).EQ.4,R4;
238      ACT,;,ATRIB(24).EQ.5,R5;
239 P1      ASSIGN, ATRIB(35)=TNOW-ATRIB(22)-ATRIB(25);
240      COLCT,ATR(35),WAITING1;
241      ACT,;,AAA;
242 R2      ASSIGN, ATRIB(35)=TNOW-ATRIB(22)-ATRIB(25);
243      COLCT,ATR(35),WAITING2;
244      ACT,;,AAA;
245 R3      ASSIGN, ATRIB(35)=TNOW-ATRIB(22)-ATRIB(25);
246      COLCT,ATR(35),WAITING3;
247      ACT,;,AAA;
248 R4      ASSIGN, ATRIB(35)=TNOW-ATRIB(22)-ATRIB(25);
249      COLCT,ATR(35),WAITING4;
250      ACT,;,AAA;
251 R5      ASSIGN, ATRIB(35)=TNOW-ATRIB(22)-ATRIB(25);
252      COLCT,ATR(35),WAITING5;
253      ACT,;,AAA;
254      ;
255 OUT     GOON,1;
256      ACT/41,;,ATRIB(24).EQ.1,F1;OUTFAM1
257      ACT/42,;,ATRIB(24).EQ.2,F2;OUTFAM2
258      ACT/43,;,ATRIB(24).EQ.3,F3;OUTFAM3
259      ACT/44,;,ATRIB(24).EQ.4,F4;OUTFAM4
260      ACT/45,;,ATRIB(24).EQ.5,F5;OUTFAM5
261 F1      COLCT,INT(23),FLOW1;
262      COLCT,INT(26),LATENESS1;
263      GOON,1;
264      ACT/46,;,ATRIB(26).GE.TNOW,XXX;EARLY1
265      ACT/47,;,ATRIB(26).LT.TNOW,XXX;LATE1
266 F2      COLCT,INT(23),FLOW2;
267      COLCT,INT(26),LATENESS2;
268      GOON,1;

```

```
269 ACT/48,,ATRIB(26).GE.TNOW,XXX:EARLY2
270 ACT/49,,ATRIB(26).LT.TNOW,XXX:LATE2
271 F3 COLCT,INT(23),FLOW3:
272 COLCT,INT(26),LATENESS3:
273 GOON,1:
274 ACT/50,,ATRIB(26).GE.TNOW,XXX:EARLY3
275 ACT/51,,ATRIB(26).LT.TNOW,XXX:LATE3
276 F4 COLCT,INT(23),FLOW4:
277 COLCT,INT(26),LATENESS4:
278 GOON,1:
279 ACT/52,,ATRIB(26).GE.TNOW,XXX:EARLY4
280 ACT/53,,ATRIB(26).LT.TNOW,XXX:LATE4
281 F5 COLCT,INT(23),FLOW5:
282 COLCT,INT(26),LATENESS5:
283 GOON,1:
284 ACT/54,,ATRIB(26).GE.TNOW,XXX:EARLY5
285 ACT/55,,ATRIB(26).LT.TNOW,XXX:LATE5
286 XXX COLCT,INT(23),FLOW TIME:
287 COLCT,INT(26),LATENESS:
288 TERMINATE,10000:
289 ENDNETWORK:
290 MONTR,CLEAR,4000:
291 MONTR,SUMRY,5000,1000:
292 FIN:
```

S L A M I I S U M M A R Y R E P O R T

SIMULATION PROJECT ATELIER

BY RACINE

DATE 7/25/1988

RUN NUMBER 1 OF 1

CURRENT TIME 0.1112E+05

STATISTICAL ARRAYS CLEARED AT TIME 0.4000E+04

STATISTICS FOR VARIABLES BASED ON OBSERVATION

	MEAN VALUE	STANDARD DEVIATION	COEFF. OF VARIATION	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBSERVATIONS
BATCH1	0.7962E+01	0.1458E+01	0.1832E+00	0.4000E+01	0.1000E+02	157
BATCH2	0.7889E+01	0.1734E+01	0.2198E+00	0.3000E+01	0.1000E+02	162
BATCH3	0.8293E+01	0.1574E+01	0.1898E+00	0.4000E+01	0.1000E+02	164
BATCH4	0.9199E+01	0.1571E+01	0.1916E+00	0.3000E+01	0.1000E+02	151
BATCH5	0.8108E+01	0.1657E+01	0.2044E+00	0.4000E+01	0.1000E+02	158
WAITING1	0.2173E+02	0.1651E+02	0.7595E+00	0.5470E-01	0.1226E+03	1250
WAITING2	0.2116E+02	0.1728E+02	0.8167E+00	0.5469E-01	0.1295E+03	1278
WAITING3	0.2142E+02	0.1671E+02	0.7801E+00	0.7031E-01	0.1174E+03	1360
WAITING4	0.2153E+02	0.1657E+02	0.7696E+00	0.7813E-01	0.1157E+03	1238
WAITING5	0.2293E+02	0.1784E+02	0.7791E+00	0.6641E-01	0.1249E+03	1281
FLOW1	0.6833E+02	0.3840E+02	0.5603E+00	0.2202E+02	0.4981E+03	1254
LATENESS1	-0.1407E+01	0.3480E+02	-0.2472E+02	-0.1114E+03	0.3587E+03	1254
FLOW2	0.6519E+02	0.4944E+02	0.7584E+00	0.1770E+02	0.6078E+03	1273
LATENESS2	-0.2731E+00	0.4687E+02	-0.1722E+03	-0.9429E+02	0.5029E+03	1273
FLOW3	0.6781E+02	0.4768E+02	0.7032E+00	0.2004E+02	0.8337E+03	1365
LATENESS3	-0.7750E+00	0.4379E+02	0.5630E+02	-0.1019E+03	0.7012E+03	1365
FLOW4	0.7552E+02	0.3897E+02	0.5525E+00	0.2436E+02	0.6074E+03	1239
LATENESS4	-0.1196E+01	0.3707E+02	-0.3099E+02	-0.8716E+02	0.5285E+03	1239
FLOW5	0.6435E+02	0.4053E+02	0.6298E+00	0.1418E+02	0.5171E+03	1275
LATENESS5	0.3021E+00	0.3775E+02	0.1249E+03	-0.9433E+02	0.4319E+03	1275
FLOW TIME	0.6725E+02	0.4339E+02	0.6453E+00	0.1418E+02	0.8337E+03	6406
LATENESS	-0.3352E+00	0.4038E+02	-0.1205E+03	-0.1114E+03	0.7012E+03	6406

FILE STATISTICS

FILE NUMBER	ASSOC LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	M1 QUEUE	2.7292	2.4040	15	2	3.0306
2	M2 QUEUE	2.7045	2.2704	12	1	3.0041
3	M3 QUEUE	3.5303	3.0983	17	4	3.9201
4	M4 QUEUE	2.6607	2.1710	13	5	2.9545
5	M5 QUEUE	2.2497	2.1958	15	2	2.4969
6	M6 QUEUE	2.9960	2.6657	16	3	3.3289
7	M7 QUEUE	2.8788	2.6807	16	0	3.2026
8	M8 QUEUE	2.6330	2.6080	17	2	2.9283
9	M9 QUEUE	2.1999	1.9879	11	4	2.4439
10	AW1 AWAIT	0.9322	1.1424	14	2	0.8717
11	AW2 AWAIT	0.9073	1.1471	15	2	0.9272
12	AW3 AWAIT	1.0087	1.2106	15	0	0.9245
13	AW4 AWAIT	0.8878	1.1293	12	1	0.8180
14	AW5 AWAIT	0.9360	1.1395	14	0	0.9444
15	QUEUE	0.0000	0.0065	9	9	0.0000

16	QUEUE	0.0000	0.0000	7	0	0.0000
17	QUEUE	0.0000	0.0057	9	0	0.0000
18	QUEUE	0.0000	0.0093	9	0	0.0000
19	QUEUE	0.0000	0.0096	9	0	0.0000
20	FILE AWAIT	3.7276	0.5107	5	3	7.0454
21	CALENDAR	11.0896	0.9909	53	12	0.1601

REGULAR ACTIVITY STATISTICS

ACTIVITY INDEX/LABEL	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTIL	CURRENT UTIL	ENTITY COUNT
1 FAM1	0.0	0.0	1	0	1262
2 FAM2	0.0	0.0	1	0	1273
3 FAM3	0.0	0.0	1	0	1360
4 FAM4	0.0	0.0	1	0	1239
5 FAM5	0.0	0.0	1	0	1273
15 FA1	0.0	0.0	1	0	3788
16 FA2	0.0	0.0	1	0	3458
17 FA3	0.0	0.0	1	0	3849
18 FA4	0.0	0.0	1	0	3846
19 FA5	0.0	0.0	1	0	3502
20 MACHAAT	0.0	0.0	1	1	792
31 GROUP1	0.0	0.0	1	0	759
32 GROUP2	0.0	0.0	1	0	744
33 GROUP3	0.0	0.0	1	0	762
34 GROUP4	0.0	0.0	1	0	778
35 GROUP5	0.0	0.0	1	0	720
36 LOTFAM1	0.0	0.0	1	0	157
37 LOTFAM2	0.0	0.0	1	0	162
38 LOTFAM3	0.0	0.0	1	0	164
39 LOTFAM4	0.0	0.0	1	0	151
40 LOTFAM5	0.0	0.0	1	0	158
41 OUTFAM1	0.0	0.0	1	0	154
42 OUTFAM2	0.0	0.0	1	0	1273
43 OUTFAM3	0.0	0.0	1	0	1365
44 OUTFAM4	0.0	0.0	1	0	1239
45 OUTFAM5	0.0	0.0	1	0	1275
46 EARLY1	0.0	0.0	1	0	747
47 LATE1	0.0	0.0	1	0	507
48 EARLY2	0.0	0.0	1	0	792
49 LATE2	0.0	0.0	1	0	481
50 EARLY3	0.0	0.0	1	0	807
51 LATE3	0.0	0.0	1	0	558
52 EARLY4	0.0	0.0	1	0	751
53 LATE4	0.0	0.0	1	0	488
54 EARLY5	0.0	0.0	1	0	737
55 LATE5	0.0	0.0	1	0	538

SERVICE ACTIVITY STATISTICS

ACTIVITY INDEX	START NODE OR ACTIVITY LABEL	SERVER CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	CURRENT UTILIZATION	AVERAGE BLOCKAGE	MAXIMUM IDLE TIME/SERVERS	MAXIMUM BUSY TIME/SERVERS	ENTITY COUNT
6	MACH1	1	0.9271	0.2902	1	0.0	5.2695	413.4961	6410
7	MACH2	1	0.9060	0.2918	1	0.0	7.9141	611.3750	6409
8	MACH3	1	0.9167	0.2763	1	0.0	7.8164	773.5977	6408
9	MACH4	1	0.9577	0.2894	1	0.0	6.6133	684.0859	6407
10	MACH5	1	0.8802	0.3247	1	0.0	8.1836	333.8867	6413
11	MACH6	1	0.9397	0.2866	1	0.0	6.7070	594.7500	6405
12	MACH7	1	0.8958	0.3055	1	0.0	6.1014	716.4609	6400
13	MACH8	1	0.9252	0.2822	1	0.0	6.5020	630.5627	6400

22	QUEUE	1	0.0000	0.0020	0	0.0	33.9023	0.0068
23	QUEUE	1	0.0000	0.0024	0	0.0	33.9023	0.0068
24	QUEUE	1	0.0000	0.0022	0	0.0	36.1484	0.0078
25	QUEUE	1	0.0000	0.0025	0	0.0	40.6875	0.0098

***RESOURCE STATISTICS**

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	MACHB	1	1.0000	0.0	1	1

RESOURCE NUMBER	RESOURCE LABEL	CURRENT AVAILABLE	AVERAGE AVAILABLE	MINIMUM AVAILABLE	MAXIMUM AVAILABLE
1	MACHB	0	0.0	0	0

***GATE STATISTICS**

GATE NUMBER	GATE LABEL	CURRENT STATUS	PCT. OF TIME OPEN
1	LOT1	CLOSED	0.0001
2	LOT2	CLOSED	0.0001
3	LOT3	CLOSED	0.0001
4	LOT4	CLOSED	0.0000
5	LOT5	CLOSED	0.0002

ÉCOLE POLYTECHNIQUE DE MONTRÉAL



3 9334 00290870 3