

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE INDUSTRIEL

PAR
SERGE LAMBERT

IMPLANTATION D'UN SYSTÈME DE PRODUCTION À VALEUR AJOUTÉE (J.A.T.) ET DE
L'INGÉNIERIE SIMULTANÉE CHEZ LES SYSTÈMES ÉLECTRONIQUES MATROX LTÉE.

MAI 1997

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

RÉSUMÉ

Le présent projet origine du désir de LES SYSTÈMES ÉLECTRONIQUES MATROX LTÉE d'améliorer son système de production. Le Juste-à-Temps et l'Ingénierie Simultanée sont les deux méthodes proposées à l'entreprise pour atteindre ce but. La recherche décrite dans cet ouvrage est divisée en quatre parties. La première partie est une mise à niveau de l'entreprise pour pouvoir fonctionner en Juste-à-Temps. Les résultats de cette partie sont présentés dans un rapport confidentiel. Le système de production étudié dans les deux prochaines parties est une ligne d'assemblage SMT. La deuxième partie traite de la flexibilité et a été réalisée par un autre étudiant, M. Bernard Cyr. La troisième partie qui est présentée dans ce mémoire tente de déterminer les règles d'ordonnancement appropriées aux différentes conditions du système de production à l'étude. Finalement, la quatrième partie porte sur l'Ingénierie Simultanée et sera traitée par un autre étudiant de maîtrise. La recherche de la partie trois est divisée en deux expériences qui ont été réalisées par simulation. Les variables étudiées dans la première expérience sont la méthode de groupement des produits en famille, la capacité des machines pour recevoir des cassettes, le facteur de similarité entre les produits, la réduction du nombre de pièces par plaquette et les règles d'ordonnancement interfamilles et intrafamille. La deuxième expérience étudie l'effet de plusieurs règles d'ordonnancement interfamilles et intrafamille en fonction des deux méthodes de groupement. Les mesures de performance sont le niveau des en-

cours moyen, les temps de passage moyens et les temps de mise en course. Un modèle de simulation AWESIM comportant des ajouts écrits en C a été utilisé pour obtenir les résultats. La validation du modèle a été faite en comparant les résultats obtenus avec ceux du système réel. L'analyse des résultats a été faite par traitement statistique. Les résultats du modèle de simulation sont très significatifs pour chacune des mesures (r -carré > 0.919). La méthode de groupement GSU* réduit de 19.9% le temps de mise en course par rapport à SDS. Les règles d'ordonnancement intrafamille SPT/PCB et interfamilles FSPT permettent de réduire le temps de passage peu importe la méthode de groupement utilisée malgré une légère augmentation du niveau des en-cours. Pour conclure, la recherche devrait être poursuivie concernant l'algorithme de formation des familles, la quantité optimale d'espace cassettes à allouer avec GSU* et l'élimination du facteur de similarité.

REMERCIEMENTS

Le Conseil de Recherches en Sciences Naturelles et Génie du Canada pour le soutien financier apporté par le biais d'une bourse à incidence industrielle.

MM. Nadim Bakhach et Bernard Cyr de Les Systèmes Électroniques Matrox Ltée pour avoir rendu cette recherche possible.

Les chercheurs Georges Abdulnour et Jocelyn Drolet de La Chaire Bombardier Sea-Doo/Ski-Doo en changement technologique de l'Université du Québec à Trois-Rivières pour leur appui durant la recherche.

TABLE DES MATIÈRES

	PAGE
RÉSUMÉ	ii
REMERCIEMENTS	iv
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	viii
LISTE DES ABRÉVIATIONS	x
CHAPITRE	
I. DÉFINITION DU PROJET.	1
1. Introduction	1
2. Problématique.	2
3. Définition du problème	3
4. Stratégie de solution.	4
II. RECHERCHE BIBLIOGRAPHIQUE	7
1. Introduction	7
2. Juste-à-Temps.	8
3. Technologie de groupe.	16
4. Ingénierie Simultanée.	18
III. PLAN DE L'EXPÉRIENCE.	27
1. Description du système	27
2. Expérimentation.	32
3. Approche de la recherche	38
IV. MODÈLE DE SIMULATION.	39
1. Description du modèle.	39

2. Validation.	40
V. ANALYSE DES RÉSULTATS	43
1. Expérience 1	43
2. Expérience 2	60
VI. CONCLUSION.	74
1. Discussion	74
2. Conclusion	76
3. Recommandations.	78
BIBLIOGRAPHIE.	80
ANNEXES	
A. DESCRIPTIONS DES PROGRAMMES	84
B. PROGRAMME DE SIMULATION	102
C. SAS EXPÉRIENCE 1.	124
D. SAS EXPÉRIENCE 2.	142

LISTE DES TABLEAUX

	PAGE
1. Règles d'ordonnancement	19
2. Caractéristiques de l'IS.	21
3. Espace cassettes des machines	28
4. Temps de mises en course.	30
5. Caractéristiques des composantes du système	30
6. Variables et niveaux de l'expérience 1.	35
7. Variables et niveaux de l'expérience 2.	37
8. Attributs de collecte de données.	40
9. Attributs de contrôle de la simulation.	41
10. Analyse de variance du niveau d'en-cours Expérience 1	47
11. Analyse de variance du temps de passage Expérience 1.	48
12. Analyse de variance du temps de mise en course Expérience 1	49
13. Analyse de Duncan Expérience 1.	50
14. Résultats sans groupement de produits	64
15. Résultats des méthodes de groupement Expérience 2	64
16. Analyse de Duncan Expérience 2.	65
17. Analyse de variance du niveau d'en-cours Expérience 2	66
18. Analyse de variance du temps de passage Expérience 2.	67
19. Analyse de variance du temps de mise en course Expérience 2	68
20. Effet des variables sur la méthode de groupement.	75

LISTE DES FIGURES

	PAGE
1. Système poussé, tiré et CONWIP.	13
2. Méthode traditionnelle vs Ingénierie Simultanée	20
3. Disposition du système de production.	28
4. Modèle de simulation.	42
5. Niveau des méthodes de groupement Expérience 1.	51
6. Niveau de capacités des cassettes Expérience 1.	52
7. Niveau du facteur de similarité Expérience 1.	53
8. Niveau de la réduction de pièces.	54
9. Niveau de l'ordonnancement interfamilles Expérience 1	54
10. Niveau de l'ordonnancement intrafamille Expérience 1.	55
11. Interaction méthodes de groupement et capacités des cassettes	56
12. Interaction méthode de groupement et facteur de similarité.	57
13. Interaction méthode de groupement et réduction de pièces.	58
14. Interaction méthode de groupement et ordonnancement interfamilles Expérience 1.	58
15. Interaction méthode de groupement et ordonnancement intrafamille Expérience 1	59
16. Niveau des méthodes de groupement Expérience 2.	69
17. Niveau de l'ordonnancement interfamilles Expérience 2	70
18. Intrafamille sans formation de famille.	71
19. Interaction méthodes de groupement et ordonnancement interfamilles Expérience 2.	72

20. Interaction méthodes de groupement et ordonnancement intrafamille Expérience 2	73
21. Programme Setfam10.	86

LISTE DES ABRÉVIATIONS

JAT	Juste-à-Temps
MRP	Planification des besoins matières
MRPLXL	Planification des besoins matières lot pour lot
CONWIP	Constant work-in-process
M-P	Matière première
Ass.	Département d'assemblage
SDS	Sequence dependent scheduling
GSU	Group set-up
SPT	Shortest processing time
FCFS	Fisrt-come, first served
MJ	Most jobs
EDD	Earliest due date
SI	Shortest imminent operation
ECON	Economic tradeoff
MW	Most work
NJNL	# jobs plus # late jobs
DK	Due date switching
IS	Ingénierie simultanée
PCB	Printed circuit board
SMT	Surface mount technology
THT	Through hole technology
Rnd	Random (Aléatoire)
FSPT	Family with the shortest processing time

CHAPITRE I

DÉFINITION DU PROJET

Introduction

Le présent projet de recherche provient d'une demande d'une entreprise oeuvrant dans l'industrie électronique. Ce projet fait parti d'un ensemble de projets qui ont été menés conjointement par plusieurs étudiants de maîtrise et chercheurs de l'université.

En effet, le secteur électronique fait face depuis quelques années à une compétition féroce dans tous les domaines, en plus de l'internationalisation des marchés. La réduction de la durée de vie des produits et des marges bénéficiaires sont les principaux effets ressentis dans l'industrie. Comme si cela ne suffisait pas, ils doivent satisfaire des clients qui veulent plus d'options (variétés). La conséquence directe étant une plus grande variété de produits à assembler et à gérer et du même coup des plus petits lots de production.

En électronique, le procédé d'assemblage de circuits imprimés consiste à placer des pièces sur un circuit imprimé (Printed Circuit Board) pour ensuite les souder. Par contre, le procédé de fabrication varie en fonction du type de technologie de pièces à assembler. Peu importe la technologie, la distinction se fait au niveau du placement des composantes sur le PCB, soit le placement en surface des pièces (Surface Mount Technology, SMT) ou le placement à travers du PCB (Through Hole

Technology, THT). Ces deux technologies sont utilisées chez Matrox. Certains produits peuvent nécessiter l'utilisation des deux technologies à différents degrés. La division de fabrication de Matrox possède deux lignes de production. La première ligne est dédiée au SMT tandis que la deuxième ligne sert pour le THT.

Problématique

Étant donné l'évolution constante et rapide des secteurs informatique et électronique, les produits conçus et fabriqués chez Matrox ont un cycle de vie réduit. Un autre point particulier à Matrox est qu'elle fabrique à la fois des produits à haut volume et plusieurs produits à bas volume. Ce phénomène a amené l'entreprise à faire des familles de produits pour minimiser les temps de mise en course. De plus, certaines composantes sont difficiles à obtenir puisqu'elles ont un délai de livraison de 6 à 12 mois. Ainsi, il devient primordial pour l'entreprise d'augmenter la flexibilité de ses lignes de production, de réduire ses stocks et le temps de passage de ses produits dans l'usine, de réduire le temps écoulé entre l'étape de conception, de production et de commercialisation en adoptant un système de gestion pro-actif pour faire face à la compétition internationale.

Pour répondre à ces exigences, deux systèmes de gestion semblent être désignés pour projeter l'entreprise vers l'amélioration continue et lui permettre de rester un chef de file dans ces domaines d'activités, ils sont le Juste-à-Temps et l'ingénierie simultanée.

Définition du problème

But

Développer un modèle générique d'implantation de nouvelles technologies par projet.

Objectif

La recherche est divisée en quatre parties, la révision du système de production, la flexibilité, l'ordonnancement et l'ingénierie simultanée.

Partie 1: Révision du système de production (Benchmarking)

Avant d'implanter un système de production Juste-à-Temps, une analyse du système de production actuel a été effectuée. Suite aux résultats de cette première étape, une stratégie d'implantation basée sur une technique de gestion de projet et d'avancement par activités selon un réseau de préséance établi a été développée.

Partie 2: La flexibilité

Cette partie veut déterminer l'effet de l'application des technologies de groupe sur l'environnement de Matrox. Elle étudiera les différentes méthodes de groupement de produit et les variables pouvant affecter celles-ci.

Partie 3: L'ordonnancement

Cette partie de la recherche étudie l'impact des différentes politiques d'ordonnancement qui se prêtent bien au Juste-à-Temps et à l'environnement de Matrox. L'étude déterminera l'effet des règles d'ordonnancement sur des variables assurant la flexibilité au système de

production modélisé. La solution retenue sera intégrée aux outils du système de planification de la production de l'entreprise.

Partie 4: L'ingénierie simultanée

La première étape pour la mise en place de l'ingénierie simultanée consistera à évaluer les besoins en information, ses utilisateurs et à quel moment dans le temps ils ont besoin des informations. Suite à cela, un modèle de système d'information tenant compte de ces éléments sera élaboré avec une stratégie d'implantation.

Stratégie de solution

Pour la réussite de cette recherche, il est important d'adopter une stratégie bien définie avant de s'attaquer au problème. La simulation, la recherche appliquée, l'expérimentation (design expérimental) et les méthodes évaluatives seront utilisées durant ce projet. Les quatre parties seront traitées de façons différentes.

Juste-à-Temps

Cette partie de la recherche se veut une mise à niveau de l'entreprise pour pouvoir fonctionner en JAT. Au tout début, un questionnaire et une méthode de travail pour caractériser l'entreprise a été développés. Le questionnaire utilisé et la méthode de travail s'y rattachant tentent de répondre aux besoins particuliers de l'entreprise. Ensuite, l'évaluation de l'entreprise a été faite à l'aide de ce questionnaire. Les éléments soulevés comme devant être corrigés sont devenus des projets à l'implantation du Juste-à-Temps pour la composante production. Selon les secteurs, ces projets seront revus à l'aide d'outils de génie industriel, comme la fiabilité, l'ergonomie, la

qualité, la réduction des temps de mise en course, et bien d'autres. Chaque projet ou activité est positionné dans un réseau de préséance selon sa priorité et sa criticité. L'étude des réponses du questionnaire révèle aussi les besoins des gestionnaires de la production. Cette première partie a déjà été complétée et fait partie d'un rapport confidentiel présenté à l'entreprise (Lambert (1996)). Plusieurs des projets découlant des recommandations ont été réalisés ou sont en évaluation.

Flexibilité

La partie sur la flexibilité et la partie sur l'ordonnancement ont été intégrées dans une seule expérience basée sur un plan factoriel d'expérimentation dans le but de cerner l'effet des variables choisies ainsi que leur interaction sur la performance du système. La description de la flexibilité et l'analyse des résultats portant sur la flexibilité sont présentées dans Cyr (1997).

Ordonnancement

Cette partie est celle qui sera traitée dans le reste de ce rapport. Le Juste-à-Temps demande beaucoup de changement et l'ordonnancement n'y échappe pas non plus. Ainsi, plusieurs propositions de systèmes de planification de la production, de suivi et de contrôle seront élaborées en tenant compte de différentes règles d'ordonnancement. Les règles premier arrivé, premier servi, le temps le plus court et les dates dues le plus tôt seront analysées avec possiblement une seconde règle pour empêcher qu'une commande ne passe trop de temps en attente. Un modèle de simulation pour les différentes propositions de systèmes a été élaboré. En premier lieu, une simulation avec les différents modèles a été faite et a servi à établir les paramètres de base communs à tous les modèles.

La ligne de production actuelle a été simulée et sert comme point de référence lors de la comparaison et de l'analyse des résultats. Ensuite, l'évaluation des systèmes pour les paramètres expérimentaux retenus a été faite par simulation et comparée au système de production actuel. Les mesures de performance étudiées sont le temps de passage, le niveau d'en-cours, l'utilisation des machines et le niveau de service. Enfin, la solution retenue sera testée en production pour valider les résultats obtenus par la simulation.

Ingénierie Simultanée

Quant à l'ingénierie simultanée, le projet consiste à déterminer quels sont les éléments ayant de l'importance pour l'entreprise. Un des éléments-clés à la réussite, après l'engagement de la haute direction, est le système d'information. Pour ce faire, les besoins en information, les utilisateurs et à quel moment ils ont besoin de cette information seront recueillis à l'aide de questionnaires et d'entrevues. À partir des données obtenues, un nouveau système d'information sera proposé. Finalement, une stratégie d'implantation répondant aux besoins de Matrox sera développée. Compte tenu de son ampleur, seulement une brève recherche bibliographique sera faite puisque ce projet sera effectué ultérieurement par un autre étudiant de maîtrise.

Le chapitre suivant présente la recherche bibliographique pour les parties de la recherches décrites ci-dessus qui seront traitées par la suite.

CHAPITRE II
RECHERCHE BIBLIOGRAPHIQUE

Introduction

Pour réussir de nos jours l'entreprise n'a plus le choix, elle doit revoir sa façon de faire les choses. La philosophie de production dite Juste-à-Temps, l'optimisation de l'ordonnancement par de nouvelles méthodes et l'Ingénierie Simultanée sont des moyens à la disposition de l'entreprise pour assurer la continuité de son succès. Ces trois notions seront approfondies tout au long de ce chapitre.

Le succès des Japonais dans tous les secteurs n'a laissé personne indifférent. Une des raisons de leur réussite est attribuée à la philosophie de production dite Juste-à-Temps. L'étude portera sur plusieurs points; les éléments-clés du Juste-à-Temps, une comparaison du Juste-à-Temps avec les autres approches de production, les facteurs de succès en Juste-à-Temps et la stratégie d'implantation. L'assignation de pièces aux machines est un problème particulier au domaine de l'électronique. De plus, l'utilisation des technologies de groupe combinées à des règles d'ordonnancement est importante pour pouvoir offrir plus de diversités de produits et en petites quantités. Le dernier outil analysé est l'Ingénierie Simultanée. L'étude portera sur son importance et ses bénéfices, ses éléments, et finalement sur la stratégie et les étapes d'implantation.

Juste-à-Temps

Le concept du Juste-à-Temps n'est pas nouveau. En effet, Henry Ford en 1926 dans son livre "Today and Tomorrow" expliquait une usine où les bateaux déchargeraient le minerai de fer pour fabriquer l'acier et que 48 heures plus tard au bout d'une ligne de production roulerait une voiture prête à être vendue. C'est exactement ce que les Japonais ont fait après la deuxième guerre mondiale et le reste de l'histoire, tout le monde la connaît.

Le Juste-à-Temps est défini par Wantuck (1989) comme étant une stratégie de production qui demande de nouvelles valeurs pour permettre de continuellement améliorer la qualité et la productivité, c'est-à-dire donner à l'entreprise une nouvelle philosophie ou culture d'entreprise. Fisher (1995), quant à lui, mentionne que c'est une philosophie de production basée sur l'élimination des pertes au niveau des ressources et l'amélioration continue de la productivité.

Dans les prochaines sections une stratégie en 7 points sera expliquée, ensuite le Juste-à-Temps sera comparé avec les autres méthodes traditionnelles, des outils de travail en Juste-à-Temps suivront et finalement une stratégie d'implantation sera présentée.

Stratégie du JAT

Cette section présente 7 principes de base pour réussir et les pièges à éviter.

Le Juste-à-Temps met l'accent sur la simplicité et la flexibilité. Pour ce faire Wantuck (1989) définit les sept principes suivants:

1. Produire ce que le client désire: Ce principe demande de la flexibilité de la part de l'entreprise. Elle doit être en mesure de répondre rapidement à une commande d'un client. Il ne sert à rien de produire selon les prévisions de vente, il est préférable d'investir dans la flexibilité manufacturière que dans des stocks qui ne seront pas vendus si les prévisions sont erronées.
2. Éliminer toutes les pertes: Les pertes sont définies comme tous les efforts superflus au niveau de l'usine, des matières et des travailleurs que ceux requis par la production. Il y a plusieurs sources de pertes, comme le non-équilibrage des tâches (machine ou opérateur), la mauvaise synchronisation des activités, les opérateurs en trop, l'excédent de capacité pour un équipement, une maintenance préventive insuffisante, les défauts et retouches et la surproduction.
3. Produire un à la fois: À long terme, l'entreprise doit chercher à produire des lots unitaires. Ce changement demande beaucoup de temps et d'autres éléments doivent être améliorés pour atteindre cet objectif. Les avantages seront la diminution des inventaires, du temps de passage des commandes et des délais de livraison.
4. Amélioration continue: L'amélioration continue est un projet sans fin. Elle consiste à se fixer des objectifs. Lorsqu'un objectif est atteint, un nouvel objectif doit être défini. C'est un processus sans fin.
5. Respect des gens: En Juste-à-Temps, tout le monde contribue et participe à l'amélioration continue. Chacun a du potentiel et doit se sentir comme faisant partie de l'entreprise.
6. Aucune contingence: L'entreprise ne doit pas ajouter des inventaires au cas où, car c'est de la perte. Ainsi, chaque ressource doit être utilisée à sa capacité requise.

7. Mettre l'accent sur le long terme: Avoir une vision long terme est très important, car le Juste-à-Temps est une stratégie qui demande du temps pour son implantation.

La qualité à la source est très importante et vise à ne produire que des bonnes pièces. Ainsi, la personne qui ajoute la valeur est entièrement responsable de la qualité de son produit.

Zygmont (1989) mentionne qu'il existe toujours des perceptions erronées sur ce qu'est le Juste-à-Temps. Une première est que certaines personnes le voient seulement comme un outil de gestion des inventaires qui prend la forme d'un kanban. Aussi, d'autres personnes disent qu'il ne peut pas fonctionner en Amérique à cause de la différence de culture qu'il y a avec celle des Japonais. Enfin, il y a des personnes qui pensent que ça touche seulement le secteur de la fabrication. Pourtant les politiques d'approvisionnement et le partenariat avec les fournisseurs sont également des facteurs de réussite.

Comparaison

Il existe deux types de systèmes de production qui sont caractérisés par la façon dont l'approvisionnement aux postes de travail est fait, le système à flux poussé et le système à flux tiré.

Le système de production le plus rencontré en Amérique est celui dit à flux poussé. Ce système fait en sorte que lorsqu'une commande est relâchée en production elle commence au premier poste et celui-ci pousse les lots de production au poste suivant, ainsi de suite jusqu'à ce qu'elle atteigne l'expédition. À la base du système, il y a un système de prévision et de planification des besoins matières (MRP) qui éclate la

demande jusqu'au plus bas niveau de la nomenclature du produit pour déterminer les ordres d'achats et de fabrication.

La méthode de planification des besoins matières lot pour lot (MRPLXL) est un cas spécial du MRP, Rees et al. (1989) explique qu'elle utilise comme taille de lot la demande et non pas une méthode basée sur le lot économique. Le système MRPLXL utilise généralement une période de planification d'une semaine.

Le Juste-à-Temps est généralement supporté par un système à flux tiré. Un système dit à flux tiré est un système où le poste en aval demande d'avoir des pièces du poste en amont lorsqu'il en a besoin. Pour approvisionner un poste, l'opérateur envoie une carte, appelée kanban au poste précédent. Les avantages du système Kanban sont une plus grande flexibilité, la simplicité, la qualité de l'information, il est peu dispendieux à utiliser et réduit le personnel requis pour faire la planification et le contrôle. De plus, Berkley et Kiran (1991) ajoutent que le JAT permet aux employés de prendre des décisions de contrôle et suivi de la production sans les délais associés à un système de planification traditionnel.

Berkley et Kiran (1991) expliquent deux types de systèmes Kanban. Le premier est un système Kanban à quantité constante. Ainsi, un poste envoie un certain nombre de kanbans de manutention au poste en amont lorsque son inventaire atteint le point de commande. Le deuxième est un système Kanban à intervalle de manutention constant. Pour celui-ci, un poste envoie à intervalle fixe au poste en amont les kanbans de manutention qu'il a utilisés pendant la période. Ces systèmes représentent des systèmes tirés périodiquement (periodic pull system).

Le système peut être à une ou deux cartes. Pour un système à une carte, la carte sert pour la production et la manutention. Tandis qu'un système à deux cartes a un kanban pour la production et un autre pour la manutention. Le kanban de production sert d'ordre de travail pour un contenant de produits finis. Le kanban de manutention sert de réquisition de matériel entre deux stations. Le système Kanban à deux cartes est un système tiré périodiquement et est plus approprié pour les ateliers qui fabriquent par lots où les opérations de manutention sont effectuées périodiquement. De plus, Monden (1993) fait mention que le nombre de kanbans par produit est fonction du temps de cycle et de l'intervalle de manutention.

Une nouvelle approche est proposée par Spearman et Zazanis (1992). Ce système possède des caractéristiques des deux systèmes précédents et est appelé le "constant work-in-process" (CONWIP). Son fonctionnement est simple. Ainsi, les nouvelles commandes sont acheminées au département d'assemblage qui ensuite signale le magasin de préparer les lots à produire. Ces lots seront ensuite poussés entre les stations. La figure 1 présente les trois stratégies énumérées plus haut.

Baker et al. (1990) ont comparé la performance du système tiré et du système poussé pour une ligne de production en série. Ils concluent qu'il n'est pas supposé y avoir de différence entre les deux systèmes dans la mesure où les deux systèmes opèrent dans les mêmes conditions.

Rees et al. (1989) , quant à eux, comparent l'utilisation de MRPLXL avec le kanban dans un atelier désorganisé au niveau de l'aménagement. Plusieurs des principes énumérés plus haut ont été utilisés dans

l'atelier sauf le regroupement technologique. Ils concluent que MRPLXL est beaucoup plus économique que les kanbans, que les inventaires sont plus bas et qu'il y a moins de mises en course des machines. La raison attribuée est qu'un système kanban, pour fonctionner, requiert d'avoir un contenant devant chaque poste dans l'atelier. De plus, le système MRPLXL se comporte mieux lorsque la demande est variable.

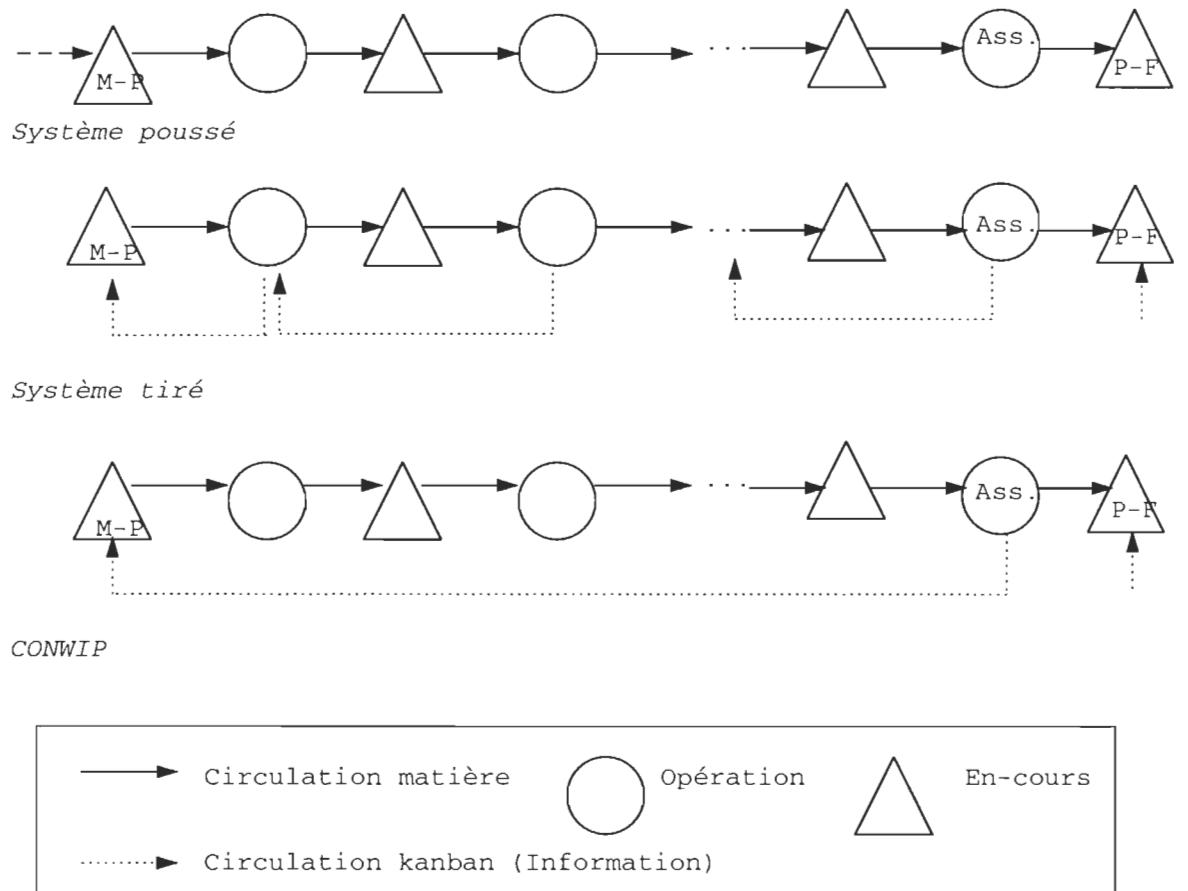


Figure 1. Système poussé, tiré et CONWIP

Co et Jacobson (1990) présentent une fonction récursive pour calculer le nombre de kanbans nécessaires à chacun des postes d'une ligne de production en série. Ils mentionnent que le fait de changer le nombre

de kanbans régulièrement en fonction de la demande est équivalent à la stratégie MRPLXL.

Spearman et Zazanis (1992) dans leur étude du CONWIP mentionne que le CONWIP est préférable au kanban dans un environnement où il y a une multitude de pièces parce qu'il n'est pas nécessaire d'avoir des en-cours pour chacune de ces pièces. Par contre, cela impose comme contrainte que le routage soit le même pour tous les produits.

Outils du JAT

Plusieurs techniques de génie industriel sont nécessaires à la mise en place du Juste-à-Temps. La première technique utilisée consiste à réduire les temps de mise en course. Aussi, il y a l'équilibrage de ligne, la révision de l'aménagement, la standardisation des produits, la qualité pour ne nommer que ceux-là. Wantuck (1989) donne les bénéfices suivant au JAT: l'augmentation de productivité, la réduction du temps de passage, la réduction des en-cours, une meilleure utilisation de la capacité de production et une plus grande flexibilité et qualité. Pour ce faire l'entreprise doit utiliser différentes techniques qui sont:

1. Cercles de qualité
2. Lignes dédiées
3. Technologie de groupe
4. Conception pour l'automatisation
5. Réduction des temps de mise en course
6. Équilibrage de ligne uniforme
7. Relation avec les fournisseurs
8. Innovation en transport

Fisher (1995), Monden (1993), Wantuck (1989), Wedderburn (1985) et plusieurs autres auteurs mentionnent comme condition critique à l'implantation du Juste-à-Temps que la demande soit nivélée pour produire tout le temps le même mixte de produits pendant une période donnée.

Stratégie d'implantation

La première étape consiste à informer les employés des changements et des raisons qui les justifient. Ensuite, des formations sur le Juste-à-Temps doivent être données au personnel clé pour qu'ils comprennent très bien ce qui sera fait par la suite. Wantuck (1989) et Fisher (1995) disent que pour vendre l'idée au reste de l'entreprise il est préférable de commencer avec le département qui a le plus grand potentiel de succès. Après avoir trouvé un département ou une unité propice, une stratégie d'implantation est requise. Hegstad (1990) présente une approche par étapes qui a été appliquée à la division Lake Stevens Instrument Division d'Hewlett-Packard. Cette entreprise produit une grande variété de produits à bas volume. La première étape d'implantation utilisée a été le CONWIP comme situation intermédiaire entre le système actuel et celui vers lequel il aspire. Ce système de transition utilise toujours les bons de travail pour pousser la matière sur le plancher et un système avec des kanbans a été mis en place pour la gestion de l'assemblage. Comme deuxième étape, l'entreprise a éliminé les bons de travail pour la matière et l'approvisionnement est maintenant fait par kanban. La dernière étape a été d'éliminer le magasin. Maintenant, la livraison est effectuée directement au point d'utilisation de la matière. Ceci a permis à H-P de réduire son temps de passage de 47 jours à 5.4 jours pour la production des appareils de mesure.

Technologie de groupe

En électronique, le procédé de fabrication utilise un nombre limité de machines. Un problème survient au moment de l'ordonnancement des commandes. Il y a plusieurs considérations qui doivent être tenues en compte comme les cadences, la variété des pièces, les pièces alternatives potentielles et les méthodes de travail. Aussi, les classes de machines sont fonction de la rapidité, de la précision, de la flexibilité, de la capacité pour charger les pièces, etc. L'ordre de passage des commandes déterminé influence directement le temps de chargement de la machine avec les composantes. Plusieurs solutions ont été proposées en utilisant le principe technologie de groupe. Ainsi, cette section traite de différentes techniques de formation de familles et des règles d'ordonnancement appropriées.

Formation de familles

Une première technique, expliquée dans Shtub et Maimon (1992), consiste à former des groupes de produits ou familles basés sur la similarité des composantes utilisées par les produits. La formation de familles vise à minimiser la somme des temps de mise en course pour augmenter l'utilisation des machines mentionnent Bhaskar et Naredran (1996). Deux techniques de formation de familles en électronique sont présentées dans la littérature: l'ordonnancement basé sur la séquence (SDS) et les mises en course groupées (GSU).

L'approche SDS (Sequence Dependent Scheduling) présentée dans Maimon et al. (1993) essaie de minimiser le changement de composants entre les produits. Hashiba et Chang (1991) expliquent que SDS consiste à regrouper en familles les produits en fonction des pièces nécessaires selon un

algorithme. Une mise en course majeure est effectuée sur les machines pour chaque nouvelle famille. Le passage entre différents produits à l'intérieur d'une famille requiert des mises en courses mineures.

La technique des mises en course groupées, appelée grouped set-up (GSU), présentée par Carmon et al. (1989) cherche à minimiser le nombre de mises en course en faisant passer tous les produits qui utilisent les composantes chargées sur une machine. Un avantage de GSU est qu'elle limite le nombre de fois qu'une pièce est chargée à une seule fois, disent Maimon et Shtub (1991). Un produit peut passer plusieurs fois sur la ligne avant d'être complètement assemblé. Ceci occasionnera une accumulation de produits partiellement assemblés en attente pour être complétés. Les en-cours générés sont ainsi plus élevés avec GSU et plus complexes à gérer, explique Carmon (1989). De plus, GSU favorise un environnement où il y a beaucoup de produits avec plusieurs pièces communes entre eux.

Règles d'ordonnancement

Plusieurs articles ont été écrits sur l'ordonnancement. Panwalkar et Iskander (1977) ont répertorié 113 règles d'ordonnancement en trois catégories. Des études de simulation ont démontré qu'il n'existe pas une règle d'ordonnancement universelle, c'est-à-dire, une règle qui donne les meilleurs résultats dans toutes les situations.

En effet, l'ordonnancement dans un système Kanban diffère d'un système de planification des besoins matières (MRP) qui recherche principalement à rencontrer les dates dues, mentionne Lee (1987). Des règles d'ordonnancement ne demandant pas de dates dues sont nécessaires pour un système Kanban. La performance des règles doit être mesurée par

des mesures autres que le retard ou l'avance d'une commande. Berkley (1993) compare deux règles couramment utilisées, le temps de production le plus court (SPT) et le premier venu, premier servi (FCFS). Une étude faite par Lee (1987) conclut que SPT/LATE, la seconde règle tenant compte du retard, est définitivement plus efficace que FCFS. Pourtant, Berkley (1993) mentionne qu'en entreprise, c'est FCFS qui est le plus utilisé. Finalement, il mentionne aussi qu'un système Kanban a des files d'attentes de capacités finies qui peuvent entraîner le blocage et leurs effets sur l'ordonnancement sont peu connus.

Après la formation des familles, les produits doivent être mis en ordre, mentionnent Hashiba et Chang (1991). L'ordonnancement peut être fait à deux niveaux, parmi les familles (interfamilles) et à l'intérieur des familles (intrafamille). Frazier (1996) présente 14 règles d'ordonnancement dont 11 sont de type interfamilles et intrafamille pour une cellule manufacturière (Tableau 1).

Ingénierie Simultanée (IS)

Un article récent dans IIE Solutions (mai 1996) indique que les compagnies oeuvrant dans le domaine de la haute technologie veulent diminuer le temps de développement d'un produit. Enfin, l'auteur mentionne que l'objectif de réduction moyen de ce délai pour les entreprises du secteur électronique est de 23.1%.

Plusieurs définitions de l'ingénierie simultanée (IS) ont été proposées au cours des dernières années, en voici deux. Broughton (1990) donne la définition suivante: "L'ingénierie simultanée tente d'optimiser la conception et la fabrication d'un produit pour obtenir une réduction

Tableau 1

Règles d'ordonnancement

Règles	Interfamilles	Intrafamille	Remarque
FCFS/FCFS	Premier arrivé, premier servi	Premier arrivé, premier servi	Arrivent toutes ensemble
MJ/SPT	Le plus de lots	Temps de passage le plus court	La plus performante selon Frazier
EDD/SI	Date due le plus tôt	Opération imminente la plus courte	Mêmes dates dues
ECON/SI	Compromis économique	Opération imminente la plus courte	
MJ/SI	Le plus de lots	Opération imminente la plus courte	
MW/SI	Le plus de travail	Opération imminente la plus courte	
MJ/Slack	Le plus de lots	La tâche qui a le moins de temps libre	Demande information sur les étapes subséquentes
MS/SPT	Temps de mise en course le plus court	Temps de passage le plus court	
NJNL/SI	Nombre de tâches + nombre de tâches en retard	Opération imminente la plus courte	Demande information sur les étapes subséquentes
DK/SI	Interchanger en fonction des dates dues	Opération imminente la plus courte	Mêmes dates dues
SPT/SPT	Temps de passage le plus court	Temps de passage le plus court	

du temps de cycle et une amélioration en qualité et coût en intégrant les activités de conception et de fabrication et en maximisant le parallélisme des tâches à accomplir". Quant à Eversheim (1990), il dit que " l'IS est une stratégie organisationnelle. L'idée est de réduire le temps de design d'un produit en planifiant simultanément le produit et la production."

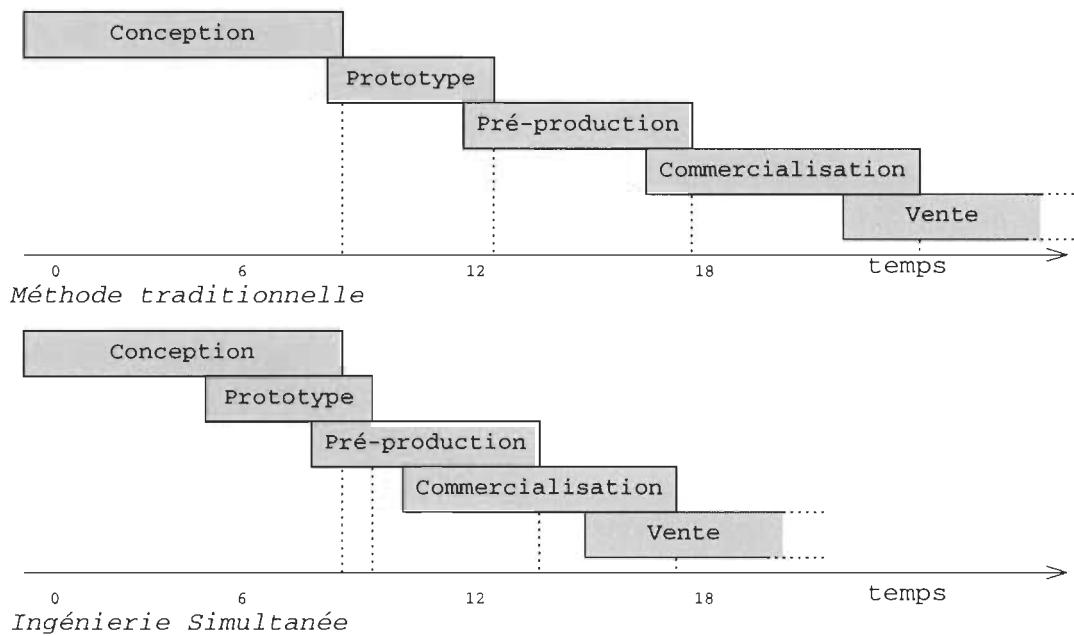


Figure 2. Méthode traditionnelle vs Ingénierie Simultanée

Importance et bénéfices de l'ingénierie simultanée

Carter (1994) mentionne que l'IS permet de réduire le temps entre la phase de la conception d'un nouveau produit et son lancement en production. La figure 2 montre la méthode traditionnelle et celle que l'IS propose sur un diagramme de Gantt. Brooks (1994) mentionne que l'IS permet d'améliorer la qualité et la conformité aux attentes du client, et apporte en plus une réduction des coûts de développement, de livraison et de support du produit. Le plus grand bénéfice de l'IS selon Carter (1994)

est un produit de meilleure qualité étant donné qu'il y a moins de modifications ou révisions apportées au produit. L'IS utilise plusieurs bases de connaissance et permet de prendre de meilleures décisions plus rapidement concernant tous les stades du cycle de vie du produit (de sa conception au déclin) fait remarquer Brooks (1994).

Éléments

L'IS fait appel à plusieurs techniques comme la qualité totale, le déploiement de la fonction qualité ou la conception pour la fabrication et l'assemblage. Brooks (1994) énumère neuf éléments qui indiquent que l'entreprise fait de l'IS. Le chapitre 21 du livre de Kusiak (1992) présente 10 caractéristiques avec un peu plus de détail que Brooks. Ces éléments donnent une bonne idée de ce que signifie faire de l'IS. Le tableau 2 montre ces caractéristiques et ce qui est requis.

Tableau 2
Caractéristiques de l'IS

No	Caractéristiques	Ce qui est requis
1	Une approche de conception "top-down" basée sur un système d'ingénierie des procédés maîtrisés	<ul style="list-style-type: none"> - Support autoritaire et participatif de la haute direction pour bâtir un consensus. - Développement d'un plan de gestion des systèmes d'ingénierie. - Un système d'information intégré par ordinateur pour permettre la gestion et le contrôle automatisés.
2	Une bonne interface avec les clients	<ul style="list-style-type: none"> - Méthode pour traduire la voix du client en un produit-clé ou caractéristiques du procédé. - Rester en contact avec le client durant l'évolution du produit.

Tableau 2 (Suite)

No	Caractéristiques	Ce qui est requis
3	Des équipes multifonctionnelles et multidisciplinaires	<ul style="list-style-type: none"> - Membres d'équipe de toutes les disciplines et fonctions, design, fabrication et support. - Tout le monde accepte l'opinion de chacun des membres. - La capacité d'analyse de tous les membres est égale ou presque égale.
4	La continuité des équipes	<ul style="list-style-type: none"> - Les équipes doivent être formées très tôt dans la phase de conception. - Le membre-clé devrait suivre le produit. - Former, accepter et donner des incitatifs aux membres qui suivent.
5	L'optimisation des pratiques d'ingénierie au niveau des caractéristiques des produits et des procédés	<ul style="list-style-type: none"> - Des méthodes pour incorporer des procédures d'optimisation de nature qualitative et quantitative. - Sélection des valeurs d'optimisation des produits et procédés-clés basées sur une analyse de sensibilité paramétrique.
6	Évaluation des design et prototypes grâce à la création d'un modèle digital du produit	<ul style="list-style-type: none"> - La méthode de conception par caractéristiques. - Standardiser la façon de définir le produit et l'échange de données.

Tableau 2 (Suite)

No	Caractéristiques	Ce qui est requis
7	Simulation de la performance du produit, de la fabrication et du support aux procédés	<ul style="list-style-type: none"> - Capacité de simuler par distribution. - Niveaux variés de fidélité de la simulation pour supporter l'évolution du produit pendant les étapes.
8	Expérimentation pour confirmer ou changer les prédictions à haut risque trouvées lors de la simulation	<ul style="list-style-type: none"> - Méthode de design expérimental pour la réduction de la variabilité des produits et procédés à haut risque. - Validation et vérification des composantes, pièces et technologies critiques.
9	Implication très tôt des sous-contractants et des vendeurs	<ul style="list-style-type: none"> - Décomposition organisationnelle pour identifier les chemins critiques et le parallélisme requis. - Acceptation de la participation des sous-contractants et des vendeurs.
10	Accent de la direction sur l'amélioration continue et les leçons apprises	<ul style="list-style-type: none"> - Méthodes pour suivre le design et d'obtention des leçons apprises. - Base de données et connaissances informatisées partagées et accessibles par les personnes-clés d'une équipe.

L'IS englobe plusieurs outils de travail. Brooks (1994) énumère ceux-ci: les équipes multidisciplinaires, le déploiement de la fonction qualité, la conception assistée par ordinateur, la réingénierie du processus d'affaires, un système d'information global, etc.

Stratégie et les étapes d'implantation

La stratégie pour l'implantation de l'IS est progressive. Par contre, pour réussir elle doit débuter avec l'engagement total de la haute direction, mentionne Brooks (1994). Un sondage dont les résultats sont présentés dans Kusiak (1992) démontre que les entreprises considèrent la caractéristique numéro un du tableau précédent comme la plus importante. De plus, Brooks (1994) dit que l'IS est adoptée par des entreprises suite à des initiatives des hauts dirigeants qui l'amènent progressivement à l'utiliser.

La stratégie proposée par Brooks (1994) est un modèle basé sur le cycle de vie. La première étape consiste à reconnaître qu'il y a un besoin de l'IS dans l'entreprise. Les étapes suivantes sont des étapes de développement de l'IS, la maturation du processus.

Ainsi, la deuxième étape constitue le premier pas en avant, c'est le projet pilote. Ici, des ressources sont allouées, les objectifs et mesures sont identifiés et communiqués, les membres de l'équipe sont choisis, un plan est développé, les outils et systèmes pour supporter l'IS sont identifiés, et finalement de nouvelles méthodes de travail sont développées et essayées.

À l'étape trois, des résultats commencent à être visibles. Le procédé est changé. Les membres ne travaillent plus de façon

traditionnelle. Il y a une diminution dans les barrières et les murs entre les fonctions et les départements. La performance dans les tâches est améliorée grâce au changement des comportements. Les processus commencent à prendre de nouvelles formes et la réponse est plus rapide.

La quatrième étape est celle de la croissance où plusieurs projets pilotes amènent le changement. Les procédés sont continuellement améliorés à l'aide des rétro-actions suite à l'implantation des mesures proposées par le projet initial. Il y a des rencontres pour revoir les projets et identifier les leçons apprises. La direction et les membres des projets croient en l'approche de l'IS et développent un plan à long terme.

La dernière étape est l'implantation totale de l'IS. L'implication organisationnelle de l'IS doit être prise en compte par la direction. De nouvelles façons de faire pour les systèmes de support doivent être définies et mises en place. L'implication des clients et des fournisseurs par la participation et la communication est maintenant ajoutée. Aussi, il y a l'ajout de programmes de formation sur les outils et techniques pour développer l'esprit d'équipe, l'évaluation et les processus efficents. L'approche de l'IS et de l'amélioration continue devient la norme pour tous les nouveaux projets.

Ces étapes reprennent les points mentionnés dans Kusiak (1992) et montrent comment atteindre une approche globale à l'entreprise en commençant par des petits projets au départ.

La revue de la littérature a permis de constater que très peu d'articles traitent de l'effet de l'ordonnancement dans l'assemblage de

circuits imprimés à l'aide de technologie de groupe. De plus, elle a démontré que l'IS est un très gros projet en lui-même.

Les prochains chapitres traiteront seulement de la portion retenue de la recherche, soit les parties deux et trois.

CHAPITRE III

PLAN DE L'EXPÉRIENCE

Description du système

Le système de production de Matrox est un système comportant plusieurs postes fonctionnant en série. La figure 3 montre la disposition des machines et des convoyeurs. L'espace pour les en-cours entre chacun des postes n'est pas égal. De plus, le transfert entre les différentes lignes est fait manuellement, à l'aide d'un chariot qui contient le lot.

La première ligne (SMT 1) comporte quatre machines principales. La première machine (SPP) sert à étendre la pâte à souder sur les tampons de la carte. La SPP est suivie d'une machine à placement rapide (MVII) qui dépose des pièces sur la carte. Ensuite, une machine de placement de précision (MPA) vient placer des pièces demandant un alignement précis. La MPA est une machine flexible. En effet, elle peut aussi placer des pièces requérant peu de précision sauf que sa vitesse n'est pas comparable à la MVII. Finalement, les cartes assemblées passent dans un four à convection (BTU) avant d'être soudées.

La seconde ligne (SMT 2) comporte seulement deux machines. La première machine (MVI) comporte deux têtes. Une tête dépose de la colle sur le PCB tandis que la seconde pose les composantes. La carte passera ensuite dans un four radiant (RTC) pour cuire la colle. La connexion

électrique entre les composantes et la carte est faite en passant sur la vague à souder (WAVE) par la suite.

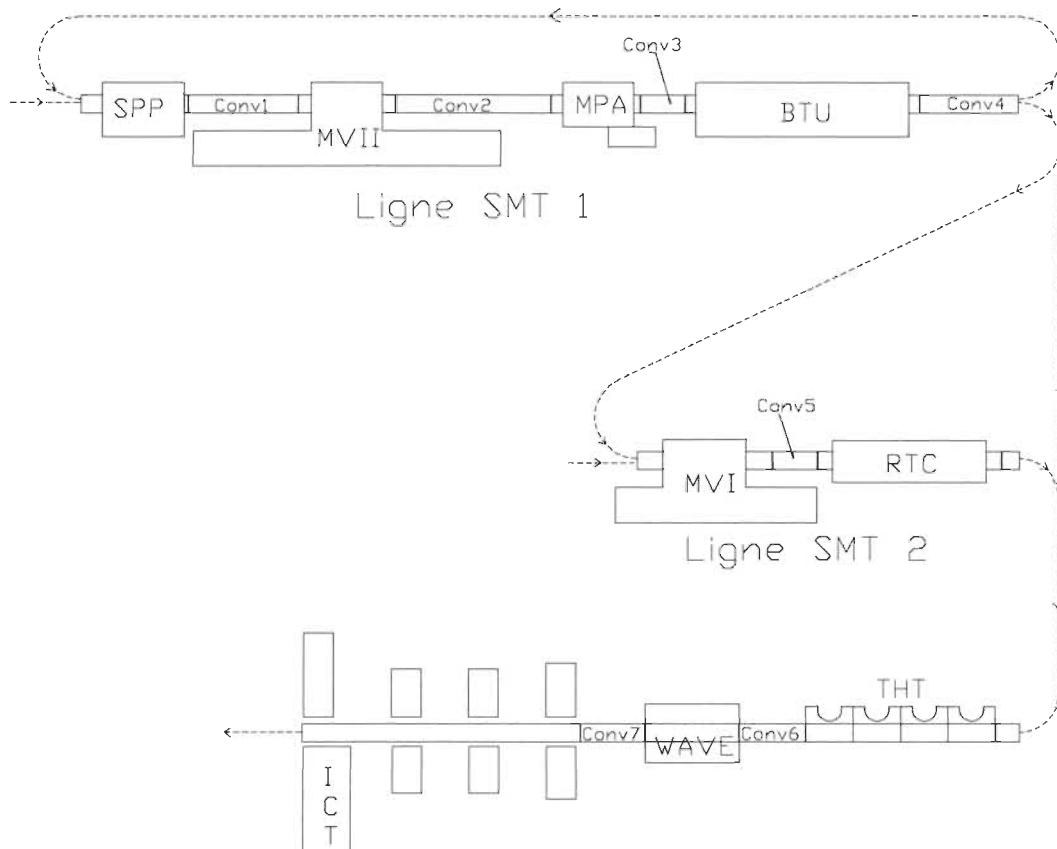


Figure 3. Disposition du système de production

Le tableau 3 montre les différents formats de composants et l'espace disponible sur chacune des machines décrites ci-dessus.

Tableau 3
Espace cassettes des machines

Types de cassettes	MPA	MVI	MVII
Rouleau	60	150	150
Tube	20	0	0
Plateau	40	0	0

Observations

Les observations concernant le fonctionnement du système de production et qui serviront dans la simulation sont les suivantes:

Principalement, Matrox possède deux lignes de production automatisée (SMT 1 et SMT 2) et une ligne d'assemblage manuel. La ligne SMT 1 est fortement sollicitée puisque presque tous les produits doivent passer par ces machines. Quant à la ligne SMT 2, sa sollicitation est plus faible, car elle sert surtout à assembler les produits avec des composantes sous la carte.

Les mises en course sont de trois types, soit le changement de famille, le changement de produit et le changement de variation. Le changement de famille demande d'enlever toutes les cassettes de pièces de la machine, les ranger, poser les cassettes requises par la nouvelle famille, s'assurer que la mise en course est bien faite, faire la mise en course du produit et finalement la mise en course de la variation. La mise en course du produit consiste à ajuster les convoyeurs, enseigner les coordonnées de la carte à la machine et faire la mise en course de la variation. La mise en course de la variation charge le programme de fabrication dans la machine et une première carte est assemblée et inspectée. Le tableau 4 montre le temps nécessaire selon le type de formation de familles utilisée.

Le tableau 5 donne les caractéristiques des composantes du système de production. Les caractéristiques données pour chacun des éléments sont sa longueur et l'espace dédié aux en-cours à l'entrée et à la sortie.

Pour certaines machines, une description plus détaillée est présentée dans les observations.

Tableau 4

Temps de mises en course

Temps de mise en course: (min.)	Aucune	SDS	GSU*
Majeur	75	75	75/35 [†]
Mineur Différent PCB	25	25	25
Mineur même PCB différente variation	15	15	15
Mineur même PCB même variation	0	0	0

[†]75 minutes pour les premiers et 35 minutes pour les autres.

Tableau 5

Caractéristiques des composantes du système

Nom	Description	Longueur	Tampon
		(PCB)	
SPP	Machine pour étendre la pâte	9.00'	1 in 1 out
Conv1	1 ^{er} convoyeur	9.25'	4
MVII	Machine à placement haute vitesse	8.00'	1 in 1 out
Conv2	2 ^{ième} convoyeur	13.00'	3
MPA	Machine à placement de précision	7.50'	1 in
Conv3	3 ^{ième} convoyeur	3.75'	1
BTU	Four à convection	20.00'	Aucun
Conv4	4 ^{ième} convoyeur	8.33	4
MVI	Machine à placement haute vitesse	11.00'	1 in 1 out
Conv5	5 ^{ième} convoyeur	3.75'	1
RTC	Four radiant	13.00'	Aucun

La machine goulot est celle qui détermine le débit de la ligne. Ainsi, il n'y aucune accumulation d'en-cours devant une machine autre que les tampons prévus.

Autres

- La mise en course est le temps total pour toutes les machines que comporte la ligne SMT.
- Certains produits requièrent un retour en arrière sur une portion de la ligne SMT. Le transfert est fait pour le lot complet.
- La planification de la production est hebdomadaire.
- Un produit est un PCB.
- Une variation de produit représente un assemblage distinct sur un même PCB.

Hypothèses

- Les commandes de la semaine ne peuvent être changées une fois que la planification est faite pour la semaine.
- Les temps de production à chaque poste sont les temps standard de l'entreprise. Le coefficient de variation de ces temps est considéré négligeable.
- Les machines ne produisent que des bonnes pièces.
- Environnement de production mixte, c'est-à-dire produits à bas volume et produits à haut volume dans une même semaine.
- Les prototypes ne sont pas assemblés durant les heures de production.
- Les pannes de machines ne sont pas considérées ici.
- Le temps de transfert des chariots est considéré nul.

Expérimentation

Le système de production de Matrox est la planification des besoins matières lots pour lots(MRPLXL). Ce système de production ne produit que les quantités requises par les commandes et ne se base pas sur la quantité du lot économique. Les commandes à assembler sont données au premier poste et elles sont ensuite poussées sur le plancher de production. L'étude portera seulement sur la ligne SMT 1 étant donné qu'elle est la plus sollicitée.

Expérience 1:

La première expérience englobe plusieurs éléments de flexibilité manufacturière et cherche à déterminer les conditions propices au système. Pour ce faire, six variables ont été retenues.

Depuis quelques années, Matrox pratique la formation de familles. Elle utilise la technique SDS et a réduit les temps de mises en course grâce à cela. L'entreprise veut connaître les avantages de GSU pour son système. Un premier problème avec la façon traditionnelle de faire GSU est la gestion des en-cours dû à la multitude de retours en arrière et de programmes des machines. Une proposition faite par M. Bernard Cyr, planificateur de la production, est de dédier la moitié des espaces cassettes sur les machines pour mettre les pièces les plus utilisées parmi les commandes de la semaine. Le reste des espaces est traité avec la technique SDS. Cette nouvelle approche a été nommée GSU* et est celle qui va être comparée à SDS.

La deuxième variable vient de l'existence sur le marché des cassettes doubles qui permettent de doubler le nombre de pièces

différentes par machine. Le coût actuel d'une cassette simple est d'environ 1 500\$ tandis que la cassette double coûte autour de 3 000\$. Son principal avantage est que les familles devraient pouvoir contenir plus de produits et ainsi diminuer le nombre de familles. Ceci se traduirait par moins de mises en course majeures.

La troisième variable tente de répondre à la question suivante: Quel est le niveau de similarité minimum pour inclure un produit dans la famille? Des essais avec différents niveaux de similarité ont été faits pour connaître son effet. En fait, il n'y pas ou très peu de changement dans la constitution des familles lorsqu'un niveau de 10% de pièces communes est imposé par rapport à aucun. Par contre, le nombre de familles commence à doubler lorsque le niveau de similarité dépasse les 30%.

Une étude interne a démontré que la liste de pièces principales de l'entreprise pourrait être réduite de près de 20% en standardisant les pièces. Par contre, cette standardisation entraînera souvent une reconception du produit. Avant de suggérer la reconception des produits, il est important de déterminer son influence sur la formation des familles et la production.

Les deux dernières variables sont liées à l'ordonnancement des commandes. Il y a deux niveaux possibles, interfamilles et intrafamille. Les règles interfamilles n'ont aucune influence sur les temps de mises en course puisque les temps de mise en course sont déterminés par l'ordre des produits à l'intérieur des familles. Par contre, elles auront une influence sur les en-cours et le temps de passage. Seulement deux règles sont étudiées ici: la première règle choisit les familles de façon

aléatoire et servira de base pour la comparaison; la seconde règle ordonnera les familles selon le temps de production le plus court (FSPT) en excluant les temps de mises en courses.

Au niveau de l'ordonnancement intrafamille, plusieurs règles peuvent être utilisées. L'utilisation de la règle premier arrivé, premier servi (FCFS) ne s'applique pas dans cette expérience puisque toutes les commandes arrivent au même moment en début de semaine. De plus, une règle en fonction de la date due ne peut être utilisée étant donné que toutes les commandes doivent être en inventaire deux semaines après leur lancement sur la ligne. Il pourrait être intéressant d'utiliser une règle basée sur le temps d'opération restant dans les autres départements et soit choisir la commande demandant le moins de temps total ou la tâche qui a le moins de temps libre pour éviter les commandes en retard. Étant données la nature des opérations et la complexité de l'affectation de la main d'œuvre, ces règles n'ont pas été retenues. La règle du temps de production le plus court (SPT) donne généralement de bons résultats. Elle a été retenue et sera comparée à la règle aléatoire. Afin d'éviter de faire plusieurs mises en course semblables, les produits ont été regroupés par PCB. De plus, Matrox procède déjà de cette manière. La première est aléatoire (RND/PCB) tandis que la seconde est selon le temps de production le plus court (SPT/PCB).

Variables

Le tableau 6 présente les variables de l'expérience 1 avec chacun des niveaux retenus.

Tableau 6
Variables et niveaux de l'expérience 1

Variables	Niveaux	
	1	2
Formation de familles	SDS (Sequence dependent set-up)	GSU* (Group set-up)
Cassette	Simple (1x)	Double (2x)
Facteur de similarité	Au moins 10%	Au moins 30%
Réduction de pièces	Aucune réduction	Réduction de 20% de la liste de pièces.
Ordre Interfamilles	Aléatoire (RND)	SPT pour la famille (FSPT) sans le temps de mise en course
Ordre Intrafamille	Aléatoire par PCB (RND/PCB)	Temps le plus court par PCB (SPT/PCB)

Mesures de performance

Les données relatives aux trois mesures de performance suivantes seront collectées au cours des simulations et seront par la suite analysées.

- Le niveau d'en-cours moyen (Cartes)
- Le temps de passage des commandes (minutes)
- Le temps de mise en course total (minutes)

Expérience 2

La deuxième expérience vient enrichir la première avec d'autres règles d'ordonnancement et mesure la performance d'un système qui ne fait

pas de formation de familles de produits. SDS et GSU* sont conservées. Cette deuxième expérience tentera de répondre à certaine question laissée par la première expérience.

Pour ce qui est des règles d'ordonnancement interfamilles, Frazier (28) présente plusieurs autres règles. Une première règle est de choisir les familles basées sur le temps minimum de mise en course (MS). Dans la première partie, les familles sont choisies sur le temps de production sans tenir compte des temps de mise en course. Ainsi, leur effet est inconnu. La règle choisie est de prendre la famille avec la somme des temps de production et de mise en course la plus petite (FSPTS).

De plus, y a-t-il un effet d'avoir groupé les produits par PCB pour les règles intrafamille? Afin de répondre à cette question, l'ordonnancement des produits aléatoirement et selon le temps de production le plus court sera évalué sans les regrouper cette fois.

Les variables suivantes ne seront pas étudiées dans la partie 2. D'abord, l'utilisation des cassettes doubles en raison de leur coût élevé et du faible gain démontré dans la première expérience. Aussi, un niveau de similarité élevé ne sera pas utilisé, car il ne favorise pas le système à l'étude en créant trop de familles. Finalement, la réduction de 20% de la liste principale de pièce se traduit par une diminution d'environ 1% du nombre de pièces utilisées dans une semaine et n'affecte que très peu le système.

Variables

Le tableau 7 présente les variables de l'expérience avec chacun des niveaux retenus de l'expérience 2.

Tableau 7
Variables et niveaux de l'expérience 2

Variables	Niveaux
Formation de familles	Aucune
	SDS (Sequence dependent set-up)
	GSU* (Group set-up)
Ordre Interfamilles	Aléatoire (RND)
	SPT pour la famille (FSPT) sans mise en course
	Temps de mise en course minimum (MS)
	SPT pour la famille avec mise en course (FSPTS)
Ordre Intrafamille	Aléatoire (RND)
	Temps le plus court (SPT)
	Aléatoire par PCB (RND/PCB)
	Temps le plus court par PCB (SPT/PCB)

Mesures de performance

Une quatrième mesure de performance a été ajoutée pour mesurer la variation entre les différentes façons de former des familles. Les mesures de performances sont:

- Le niveau d'en-cours moyen (Cartes)
- Le temps de passage des commandes (minutes)
- Le temps de mise en course (minutes)
- Le ratio temps de mise en course/temps de production et de mises en course

Approche de la recherche

Un modèle de simulation global a été construit à l'aide du logiciel de simulation AweSim. Des fonctions en langage C ont été ajoutées dans le modèle de simulation. Elles servent à la formation des familles et à trouver les structures de produits, entre autres.

L'effet des différentes variables sur les mesures de performance sera étudié par l'analyse de variance pour chacune des expériences. Le niveau de signification (α) pour les expériences est fixé à 5%.

Expérience 1:

64 (2^6) modèles de simulation \times 5 répétitions
= 320 mesures.

Expérience 2:

36 ($2^1 \times 4^2 + 1^1 \times 4^1$) modèles de simulation \times 5 répétitions
= 180 mesures.

CHAPITRE IV

MODÈLE DE SIMULATION

Description du modèle

Le modèle de simulation est écrit en langage SLAMII du logiciel Awesim. Il a été choisi à cause de sa facilité de programmation, sa possibilité d'ajouter des segments de code en complément au réseau et la qualité des générateurs d'aléas de plusieurs types de distributions de probabilité.

La simulation génère une seule entité au début et termine lorsque l'horloge a atteint 500 000 minutes. Ce temps représente environ quatre années de production lorsqu'il y a de la formation de familles sinon il est d'environ deux années. L'entité boucle à l'intérieur de la simulation pour la semaine en cours de production. L'information sur le produit à assembler est obtenue à l'aide de la fonction Userf(2). Lorsque toutes les commandes de la semaine ont été assemblées l'entité retourne dans la fonction Userf(1) pour générer une nouvelle semaine. Cette fonction crée des commandes, attribue une taille de lots, calcule le temps de production, génère les familles de produits et ordonne les produits et les familles. Le premier noeud ASSIGN du réseau SLAMII sert à déterminer les paramètres de l'expérience. Le tableau 8 donne la description des différents attributs utilisés pour la collecte de données et les unités. Le tableau 9 donne les attributs utilisés dans la simulation pour déterminer quelle expérience faire. L'annexe 1 donne la description de

chacun des programmes et l'annexe 2 donne le modèle de simulation et la fonction Userf. La figure 4 montre le fonctionnement de la simulation et l'intégration des fonctions C.

Tableau 8

Attributs de collecte de données

Attributs	Description	Unité
ATRIB[1]	Nombre de lots	
ATRIB[2]	Taille du lot	Cartes
ATRIB[3]	Temps de mise en course du lot	Minutes
ATRIB[4]	Temps d'opérations	Minutes
ATRIB[12]	En-cours en début de semaine	Cartes
ATRIB[13]	Mise en course totale de la semaine	Minutes
ATRIB[14]	Temps de passage	Minutes
ATRIB[15]	<u>Ratio mise en course</u> (mise en course + production)	

Validation

La validation des données du système a été faite à l'aide de la fenêtre interactive d'environnement qui permet de suivre en temps réel les différentes variables de la simulation. Chacun des événements pour deux semaines de production a été suivi pas à pas pour chaque variable (ATRIB[]) et mesure de performances (COLCT) afin de valider le modèle et les ajouts en langage C. Ceci a été fait pour chaque expérience. Le système modélisé est un système à capacité infinie, c'est-à-dire que lors de la planification aucune vérification de la demande est faite avec la capacité disponible. Ainsi, il n'y aura pas d'accumulation d'encours à la

fin d'une semaine. De plus, aucune étude pour déterminer à quel moment le régime permanent débute est nécessaire pour ce genre de système.

Tableau 9
Attributs de contrôle de la simulation

Attributs	Description	Valeur	Niveau
ATTRIB[5]	Formation de familles	0	Aucune
		1	SDS
		2	GSU*
ATTRIB[6]	Capacité des cassettes	1	1x
		2	2x
ATTRIB[7]	Facteur de similarité	1	10%
		2	30%
ATTRIB[8]	Réduction de pièces	1	Aucune
		2	20%
ATTRIB[9]	Groupement Intra	1	Non
		2	Par PCB
ATTRIB[10]	Ordre Interfamilles	1	RND
		2	FSPT
		3	MS
		4	FSPTS
ATTRIB[11]	Ordre Intrafamille	1	RND
		2	FSPT

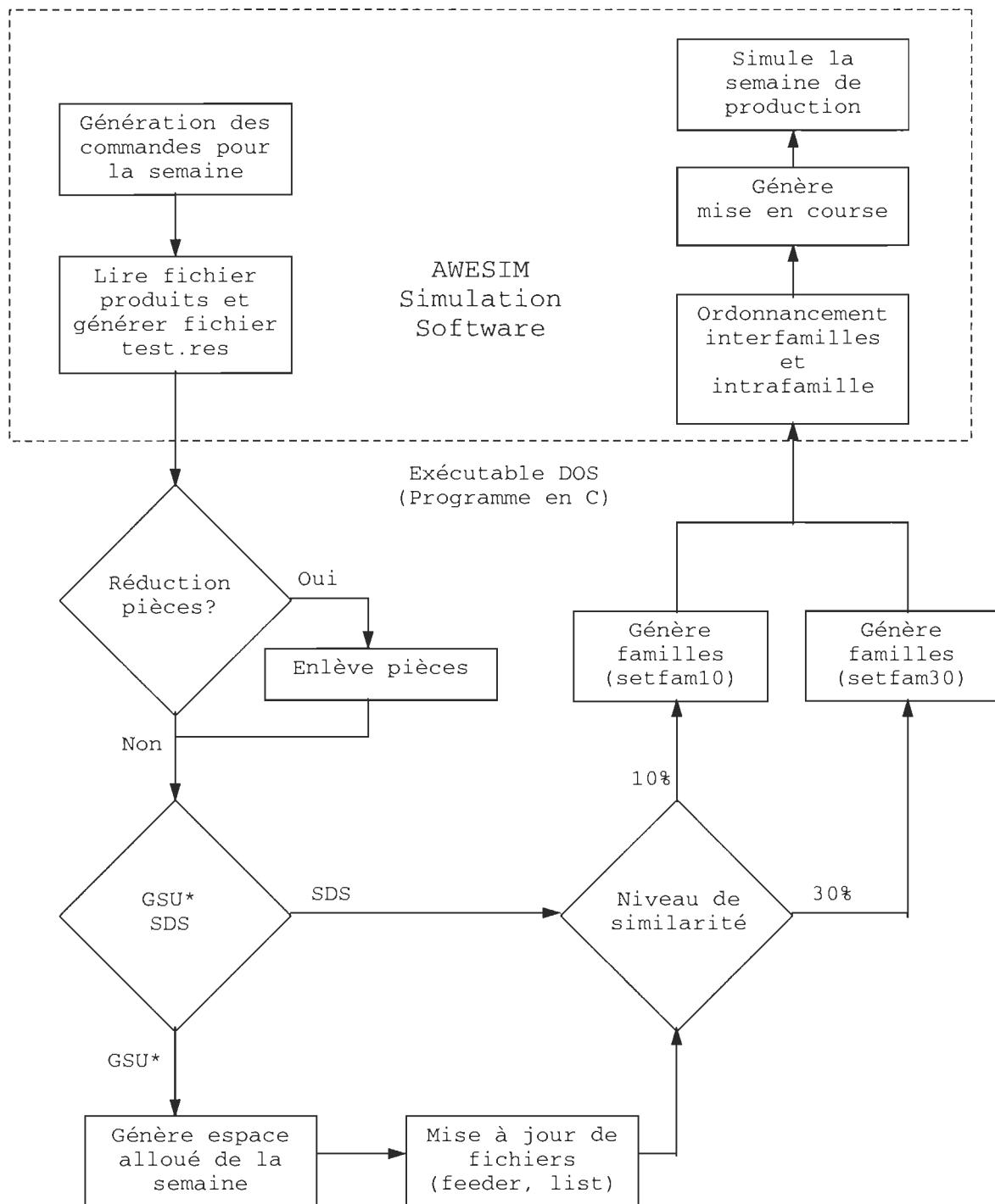


Figure 4. Modèle de simulation

CHAPITRE V

ANALYSE DES RÉSULTATS

Expérience 1

Les tableaux 10 à 12 présentent l'analyse de variance pour la première expérience. Les résultats sont très significatifs pour les trois mesures de performance puisque leurs r-carrés respectifs sont très élevés. Le tableau 13 présente l'analyse de similarité à l'aide du test de Duncan des niveaux des variables pour les trois mesures de performance.

Méthode de groupement

Les résultats pour la méthode de groupement sont significatifs pour les trois mesures de performance. En effet, la méthode de groupement GSU* a pour effet d'augmenter légèrement le niveau moyen des en-cours d'environ 14 cartes, soit 2%, par rapport à SDS. De plus, le temps de passage moyen des commandes est significativement influencé par la méthode de groupement. Ainsi, GSU* diminue de près de 78 minutes ou 6.4% le temps de passage moyen des commandes. Le plus gros gain est au niveau du temps de mise en course. En effet, GSU* le diminue de près de 200 minutes c'est-à-dire 19.9%. (voir figure 5).

Capacité des cassettes

L'utilisation des cassettes doubles réduit d'environ 92 minutes, soit 10%, le temps nécessaire pour les mises en course. La figure 6

montre que le gain est moindre avec 4% pour le temps de passage. L'effet sur les en-cours est non significatif puisque la taille de lot n'est pas affectée par la réduction de pièces.

Facteur de similarité

L'utilisation d'un facteur de similarité a pour effet de fixer un niveau de similarité entre les produits lors de la formation de familles. Le fait de le fixer à un niveau plus élevé force la création de plus de familles. Elle n'entraîne aucune réduction statistiquement significative des en-cours puisque la taille des lots n'est pas affectée. La figure 7 montre que plus son niveau est élevé plus il y aura de familles, donc plus de mises en course majeures. Ceci a pour effet d'augmenter le temps moyen de mise en course de plus de 16.5% ou 132 minutes et le temps de passage de 8% ou 92 minutes. L'augmentation du temps est équivalente à l'ajout de 2 mises en course majeures et l'élimination de 2 mises en course mineures environ.

Réduction de pièces

La réduction de pièces est seulement significative pour le temps de mise en course bien que la diminution soit de seulement 8 minutes ce qui représente moins de 1%. (figure 8).

Ordonnancement interfamilles

L'ordre attribué aux familles n'a aucun effet statistiquement significatif sur le temps de mise en course. Cela est bien normal puisque les temps de mise en course sont indépendants de l'ordre des familles. La figure 9 montre que l'utilisation de la règle d'ordonnancement des familles FSPT permet de réduire le temps de passage d'environ 135 minutes

soit 10.8%. Par contre, il augmente le niveau des en-cours de 139 cartes (22%) par rapport à Rnd.

Ordonnancement intrafamille

L'ordonnancement à l'intérieur de la famille est significatif pour les trois mesures de performance. La règle SPT/PCB augmente de 30 cartes ou de 4% le niveau des en-cours comparativement à Rnd. Aussi, elle réduit le temps de passage de 32 minutes ou 2.7%. Les temps de mise en course sont dépendants de l'ordre des produits. Quant au temps de mise en course, il augmente de 10 minutes c'est-à-dire près de 1.2% avec l'utilisation de SPT/PCB. (figure 10.)

Interactions

L'interaction entre la méthode de groupement et la capacité des cassettes est présentée à la figure 11. L'utilisation de cassettes simples ou doubles avec GSU* n'affecte pas beaucoup la performance du système.

La méthode de groupement GSU* semble moins affectée par le facteur de similarité au niveau du temps de passage et du temps de mise en course. (figure 12.)

La figure 13 montre l'interaction entre la réduction de pièces et la méthode de regroupement. Le temps de mise en course est influencé significativement par cette interaction selon l'analyse de variance. Par contre, l'analyse graphique montre une faible diminution du temps de mise en course due à la réduction de pièces.

La règle d'ordonnancement interfamilles SPT donne de meilleurs résultats avec GSU* par rapport à SDS. Elle permet une réduction du temps de passage moyen de 232 minutes (18.8%) sans avoir d'effet sur le temps de mise en course. Par contre, SPT entraîne une augmentation des en-cours de plus de 45 cartes soit de 6%. Le passage de la technique de groupement SDS à GSU* combiné à la règle d'ordonnancement Rnd diminue un peu le niveau d'inventaire (15 cartes ou 2.3%). Malheureusement, le temps de passage augmente de près de 80 minutes ou 6.25% avec GSU*. (figure 14.)

Les résultats de l'interaction entre la méthode de groupement et les règles d'ordonnancement intrafamille sont significatifs pour les trois mesures de performance. Le comportement des règles d'ordonnancement intrafamille donne un écart moins grand avec GSU* lorsque comparé avec la méthode de groupement comme le montre la figure 15. SPT/PCB donne un temps de passage plus faible que Rnd/PCB dans les deux cas. L'utilisation de GSU* réduit de près de 200 minutes (20.6%) le temps de mise en course pour les deux règles par rapport à SDS. Rnd/PCB donne un temps relativement plus faible que SPT/PCB.

Tableau 10

Analyse de variance du niveau d'en-cours Expérience 1

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	21	1825585.6125	86932.6482	189.26	0.0001
Error	298	136883.5009	459.3406		
Corrected Total	319	1962469.1134			
	R-Square	C.V.	Root MSE		WIP Mean
	0.930249	3.047973	21.432233		703.16345
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	15966.5367	15966.5367	34.76	0.0001
FEEDER	1	447.5276	447.5276	0.97	0.3244
RHO	1	709.3429	709.3429	1.54	0.2150
REDUCT	1	303.0351	303.0351	0.66	0.4173
INTERF	1	1546783.2722	1546783.2722	3367.40	0.0001
INTRAF	1	71230.8551	71230.8551	155.07	0.0001
METHODE*FEEDER	1	13015.2208	13015.2208	28.33	0.0001
METHODE*RHO	1	16883.9765	16883.9765	36.76	0.0001
METHODE*REDUCT	1	129.8530	129.8530	0.28	0.5953
METHODE*INTERF	1	75546.1045	75546.1045	164.47	0.0001
METHODE*INTRAF	1	11532.2977	11532.2977	25.11	0.0001
FEEDER*RHO	1	443.8241	443.8241	0.97	0.3264
FEEDER*REDUCT	1	0.2584	0.2584	0.00	0.9811
FEEDER*INTERF	1	49215.0691	49215.0691	107.14	0.0001
FEEDER*INTRAF	1	6841.4876	6841.4876	14.89	0.0001
RHO*REDUCT	1	9.9061	9.9061	0.02	0.8833
RHO*INTERF	1	15192.2813	15192.2813	33.07	0.0001
RHO*INTRAF	1	1189.6251	1189.6251	2.59	0.1086
REDUCT*INTERF	1	141.8491	141.8491	0.31	0.5788
REDUCT*INTRAF	1	2.7086	2.7086	0.01	0.9388
INTERF*INTRAF	1	0.5810	0.5810	0.00	0.9717

Tableau 11
Analyse de variance du temps de passage Expérience 1

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	21	5724563.7937	272598.2759	242.38	0.0001
Error	298	335155.5033	1124.6829		
Corrected Total	319	6059719.2970			
	R-Square	C.V.	Root MSE		TPAS Mean
	0.944691	2.832439	33.536292		1184.0076
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	486983.3607	486983.3607	433.00	0.0001
FEEDER	1	192781.9779	192781.9779	171.41	0.0001
RHO	1	672491.9568	672491.9568	597.94	0.0001
REDUCT	1	80.1674	80.1674	0.07	0.7897
INTERF	1	1471582.2274	1471582.2274	1308.44	0.0001
INTRAF	1	81722.7475	81722.7475	72.66	0.0001
METHODE*FEEDER	1	188257.5456	188257.5456	167.39	0.0001
METHODE*RHO	1	340117.1842	340117.1842	302.41	0.0001
METHODE*REDUCT	1	2721.0210	2721.0210	2.42	0.1209
METHODE*INTERF	1	1932058.7455	1932058.7455	1717.87	0.0001
METHODE*INTRAF	1	8589.4489	8589.4489	7.64	0.0061
FEEDER*RHO	1	192.1998	192.1998	0.17	0.6796
FEEDER*REDUCT	1	17.2972	17.2972	0.02	0.9014
FEEDER*INTERF	1	310172.2675	310172.2675	275.79	0.0001
FEEDER*INTRAF	1	5644.6977	5644.6977	5.02	0.0258
RHO*REDUCT	1	888.8240	888.8240	0.79	0.3747
RHO*INTERF	1	27031.5571	27031.5571	24.03	0.0001
RHO*INTRAF	1	1071.0597	1071.0597	0.95	0.3299
REDUCT*INTERF	1	2138.9189	2138.9189	1.90	0.1689
REDUCT*INTRAF	1	5.2438	5.2438	0.00	0.9456
INTERF*INTRAF	1	15.3450	15.3450	0.01	0.9071

Tableau 12

Analyse de variance du temps de mise en course Expérience 1

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	21	6055265.8081	288345.9909	13771.94	0.0001
Error	298	6239.2885	20.9372		
Corrected Total	319	6061505.0966			
		R-Square	C.V.	Root MSE	SETUPMean
		0.998971	0.528629	4.5757196	865.58309
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	2903607.5340	2903607.5340	99999.99	0.0001
FEEDER	1	679831.0254	679831.0254	32469.99	0.0001
RHO	1	1400950.5578	1400950.5578	66912.00	0.0001
REDUCT	1	4170.0163	4170.0163	199.17	0.0001
INTERF	1	0.0000	0.0000	0.00	0.9995
INTRAF	1	7948.2948	7948.2948	379.63	0.0001
METHODE*FEEDER	1	485586.7921	485586.7921	23192.53	0.0001
METHODE*RHO	1	548160.8684	548160.8684	26181.18	0.0001
METHODE*REDUCT	1	2662.8050	2662.8050	127.18	0.0001
METHODE*INTERF	1	0.0000	0.0000	0.00	0.9995
METHODE*INTRAF	1	2611.9455	2611.9455	124.75	0.0001
FEEDER*RHO	1	16434.8763	16434.8763	784.96	0.0001
FEEDER*REDUCT	1	114.3795	114.3795	5.46	0.0201
FEEDER*INTERF	1	0.0000	0.0000	0.00	0.9995
FEEDER*INTRAF	1	817.1507	817.1507	39.03	0.0001
RHO*REDUCT	1	1361.0148	1361.0148	65.00	0.0001
RHO*INTERF	1	0.0000	0.0000	0.00	0.9995
RHO*INTRAF	1	1007.5606	1007.5606	48.12	0.0001
REDUCT*INTERF	1	0.0000	0.0000	0.00	0.9995
REDUCT*INTRAF	1	0.9870	0.9870	0.05	0.8283
INTERF*INTRAF	1	0.0000	0.0000	0.00	0.9995

Tableau 13
Analyse de Duncan Expérience 1

Mesure de performance	En-cours	Temps de passage	Temps de mise en course
Variable	Niveau	Moyenne	Moyenne
Alpha = 0.05	DF = 298		
Carré moyen	459.3406	1124.683	20.93721
Écart critique	4.716	7.379	1.007
Méthode	SDS	696.100	1223.018
	GSU*	710.227	1144.997
Cassettes	1x	704.346†	1208.552
	2x	701.981†	1159.463
Similarité	10%	704.652†	1138.165
	30%	701.675†	1229.850
Réduction	Aucune	704.137†	1184.508†
de pièces	Red20%	702.190†	1183.507†
Interfamilles	Rnd	633.639	1251.821
	SPT	772.688	1116.194
Intrrafamille	Rnd/PCB	688.244	1199.988
	SPT/PCB	718.083	1168.027
			870.5669

† Ne sont pas significativement différents.

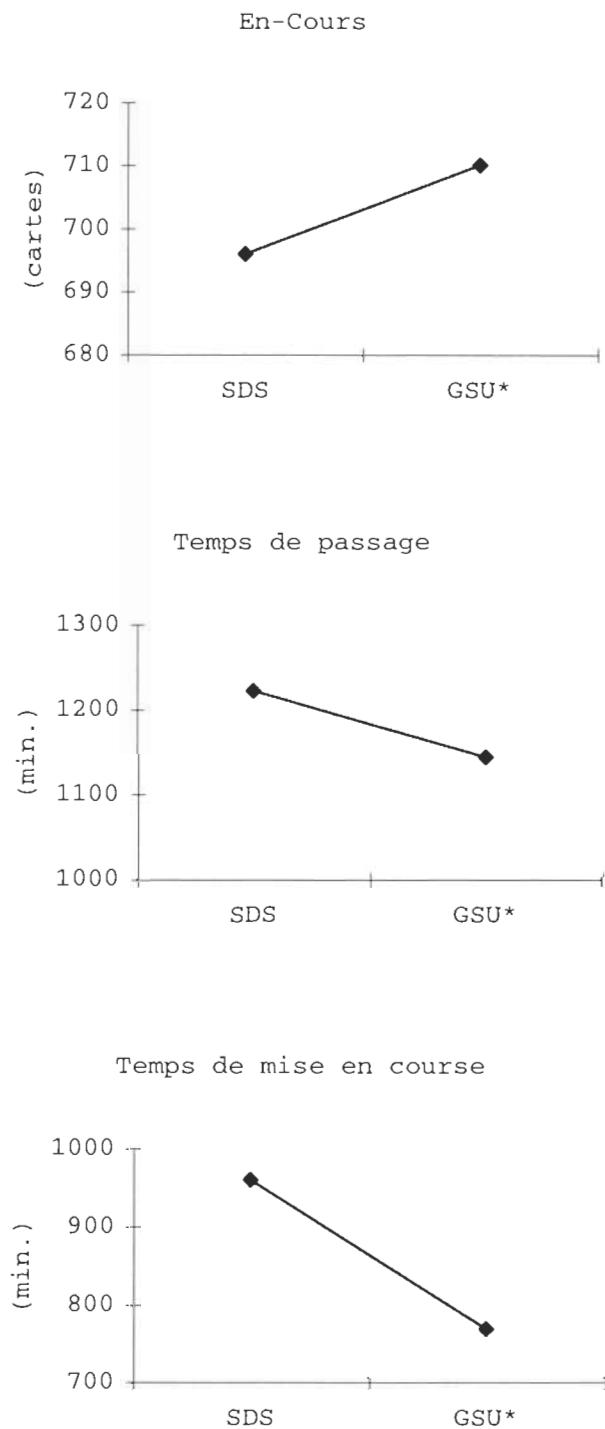


Figure 5. Niveau des méthodes de groupement Expérience 1

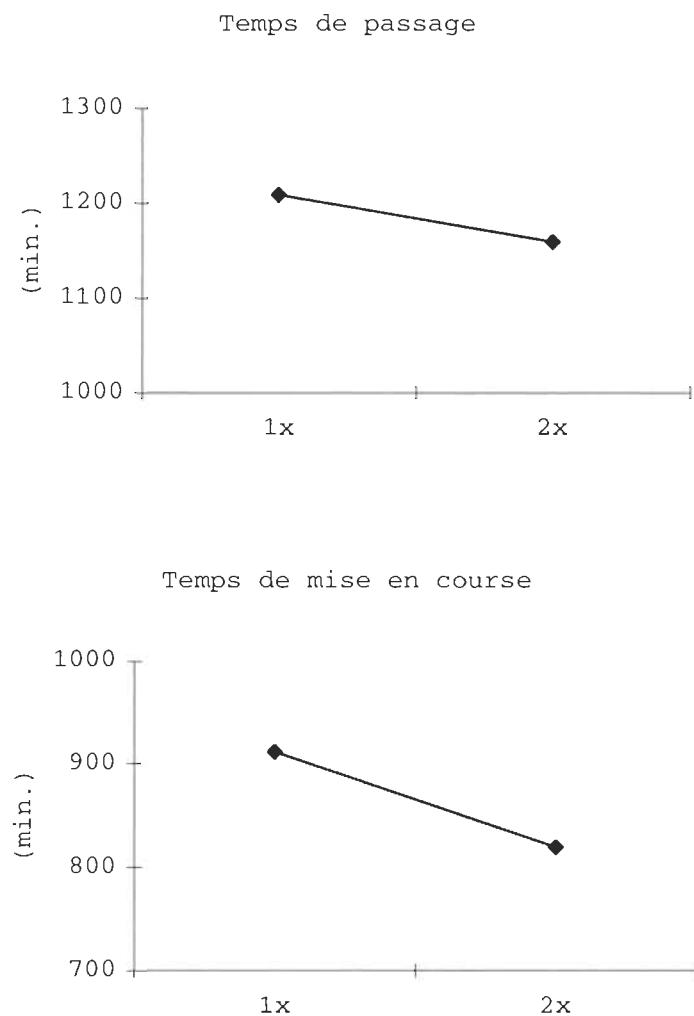


Figure 6. Niveau de capacité des cassettes Expérience 1

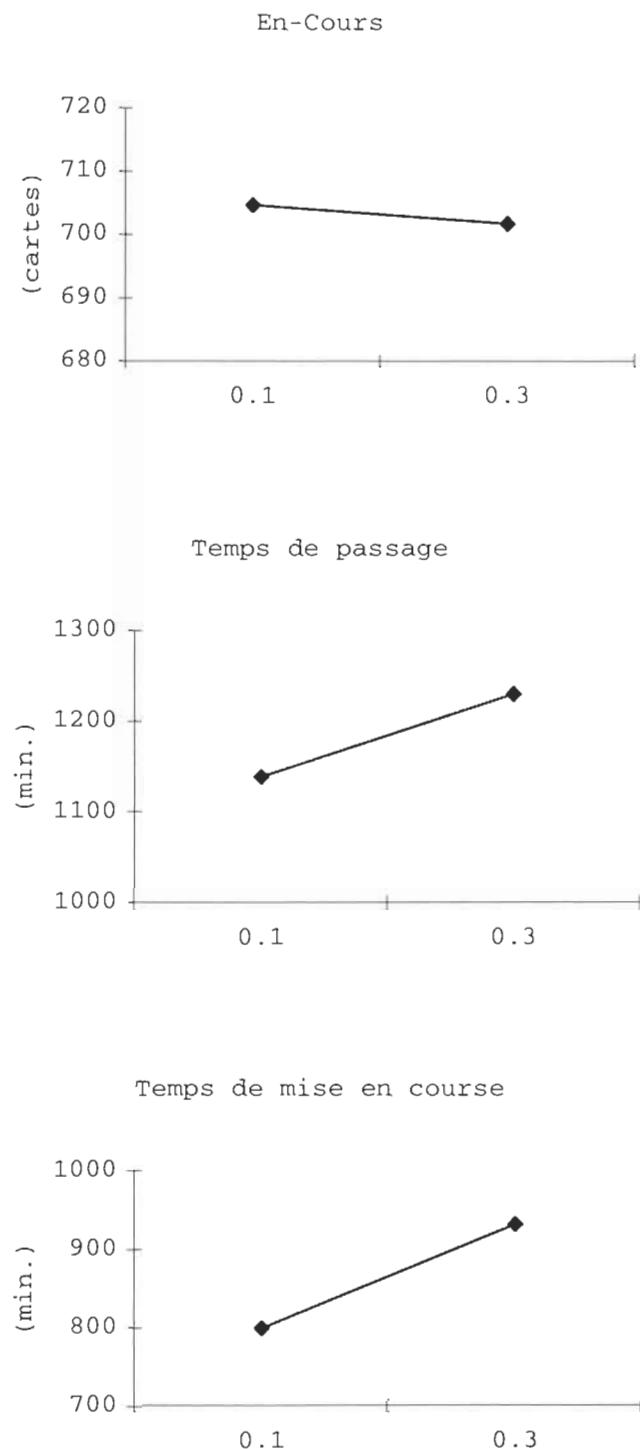


Figure 7. Niveau du facteur de similarité Expérience 1

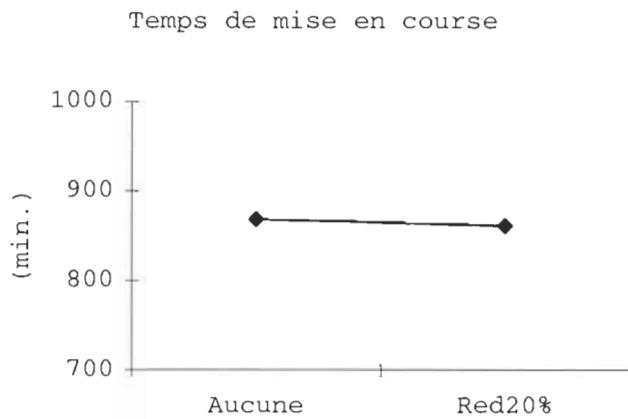


Figure 8. Niveau de la réduction de pièces

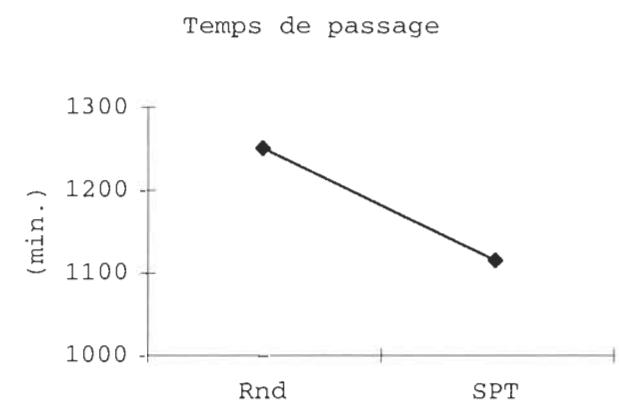
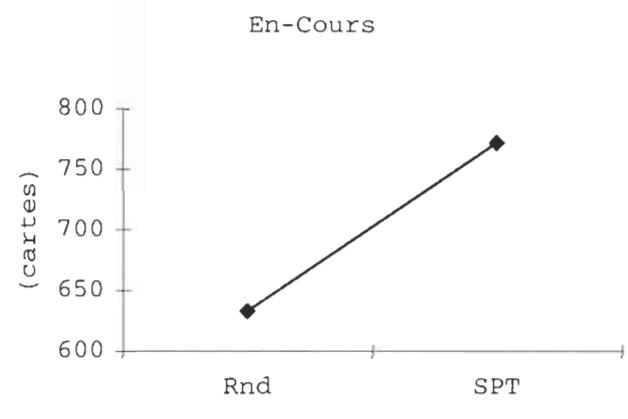


Figure 9. Niveau de l'ordonnancement interfamilles Expérience 1

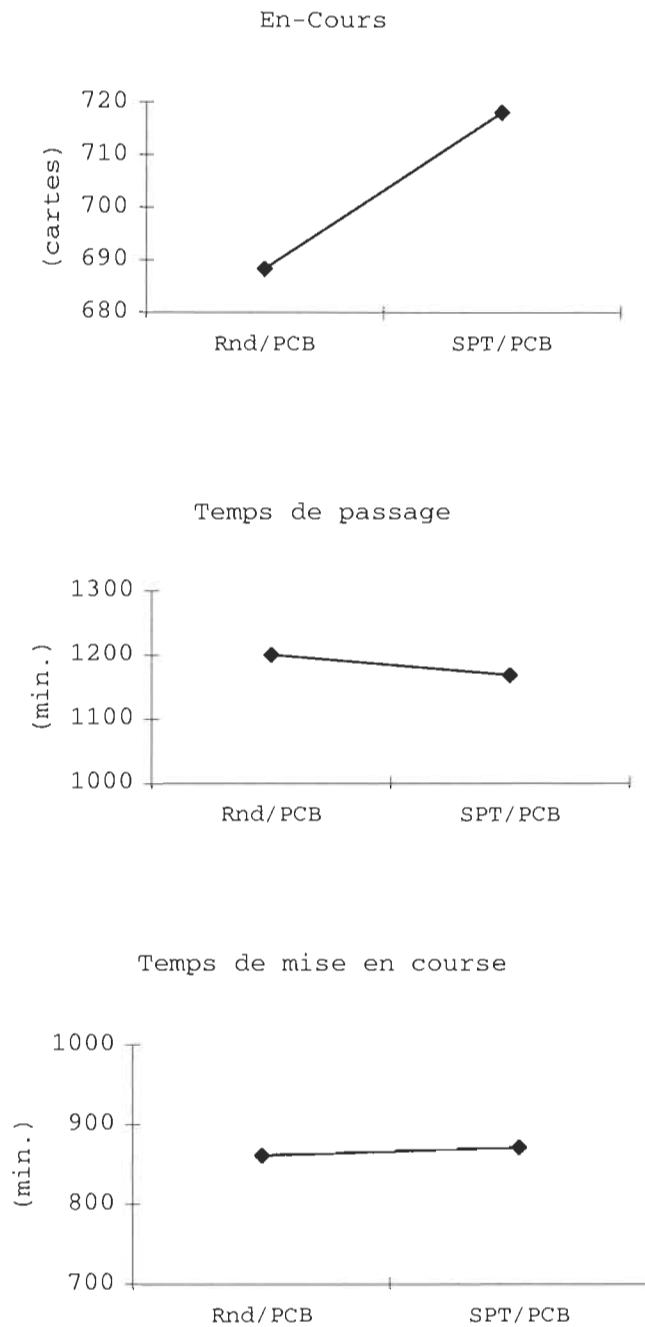


Figure 10. Niveau de l'ordonnancement intrafamille Expérience 1

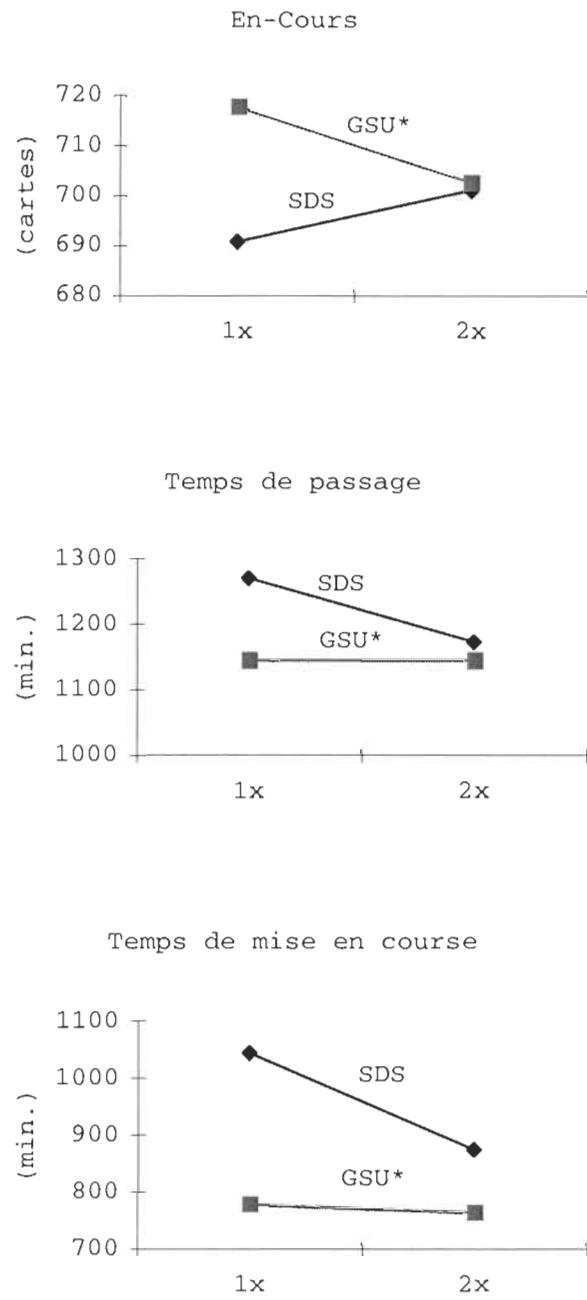


Figure 11. Interaction méthodes de groupement et capacités des cassettes

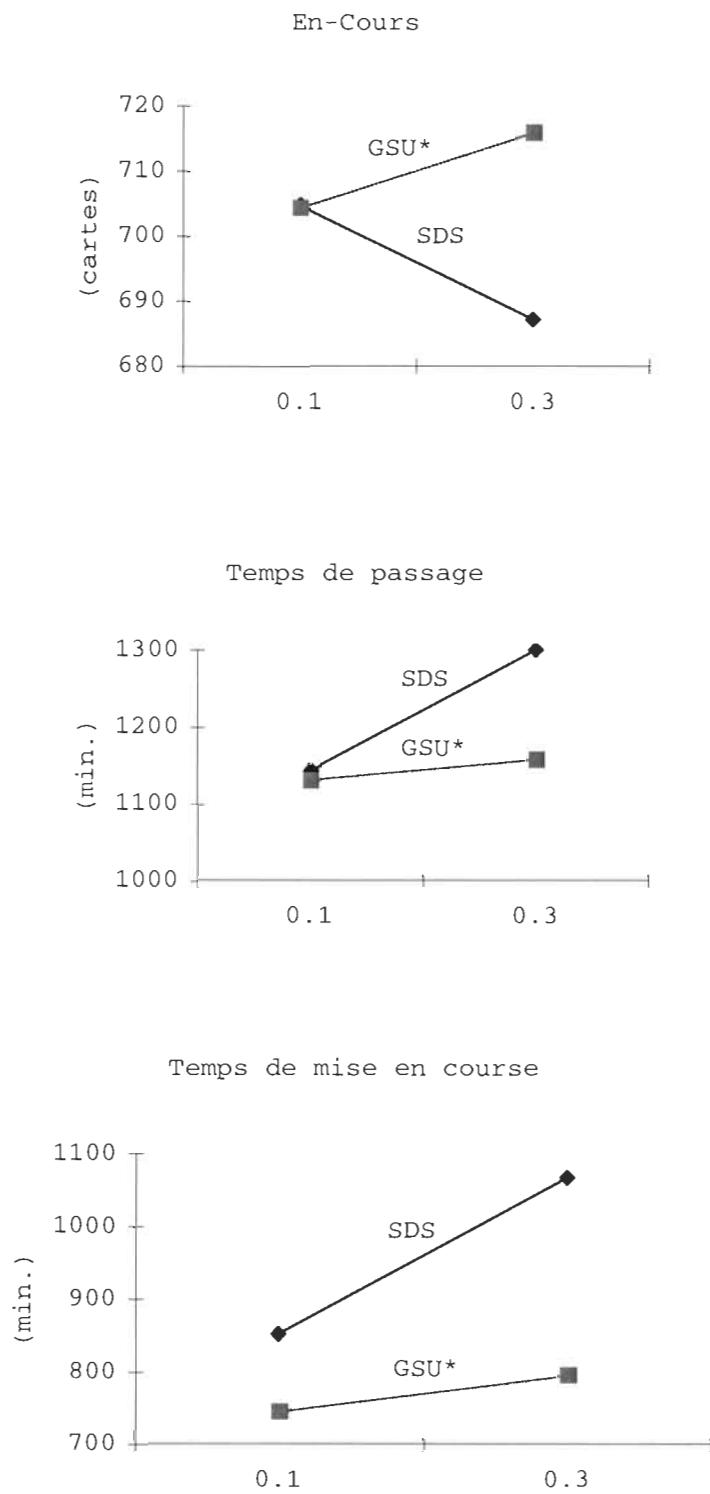


Figure 12. Interaction méthode de groupement et facteur de similarité

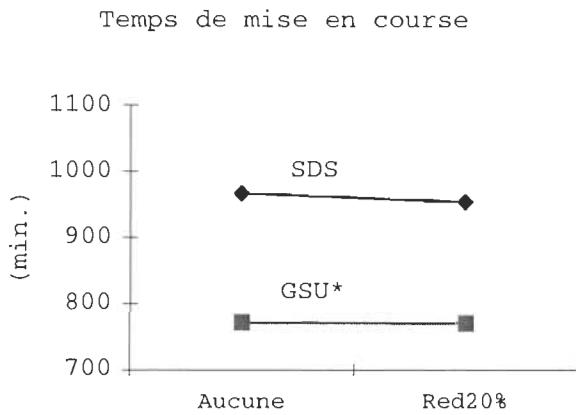


Figure 13. Interaction méthode de groupement et réduction de pièces

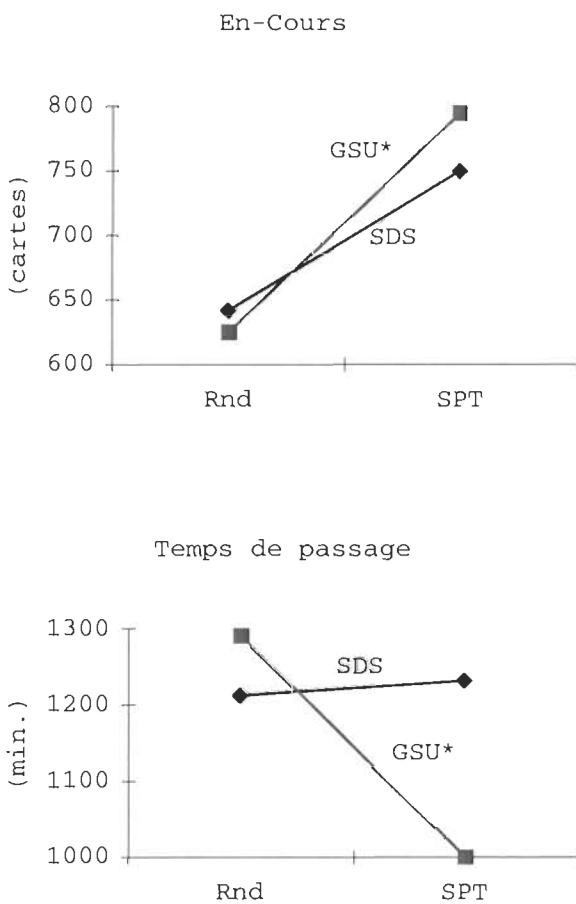


Figure 14. Interaction méthode de groupement et ordonnancement inter famille Expérience 1

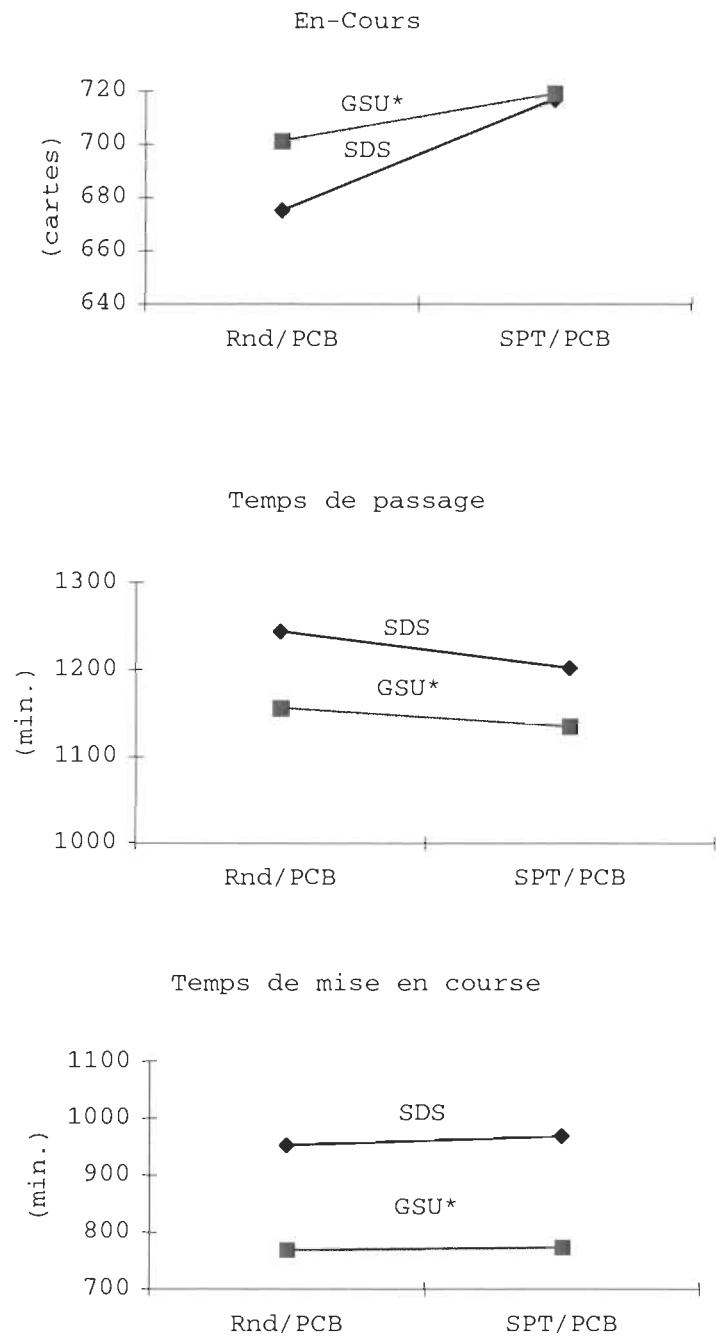


Figure 15. Interaction méthode de groupement et ordonnancement intra famille Expérience 1

Expérience 2

Les tableaux 14 et 15 présentent les résultats lorsqu'il n'y pas de formation de familles en fonction des règles d'ordonnancement et la comparaison entre les deux méthodes de groupement ou ne pas en faire. Le tableau 16 présente l'analyse de similarité à l'aide du test de Duncan des niveaux des variables pour les trois mesures de performance. Les tableaux 17 à 19 présentent l'analyse de la variance pour chacune des mesures de performances. Les R-Carrés des trois mesures sont très significatifs. Pour le temps de mise en course, le modèle explique 98.8% des variations. Le niveau d'en-cours est influencé par les trois variables de l'expérience et par l'interaction de la méthode de groupement et de la règle d'ordonnancement interfamilles.

Formation de familles

Le tableau 15 montre l'effet des différentes façons de grouper les produits sur les en-cours, le temps de passage, le temps de mise en course et le ratio du temps de mise en course sur le temps de production. La figure 16 montre clairement qu'il y a un gain grâce à la formation de familles. Il y a une faible diminution au niveau des en-cours suite à l'utilisation des familles de produits, soit 5.8% pour SDS et 4.7% pour GSU*. La formation de familles avec la méthode SDS permet une réduction de 1110 minutes (52%) des temps de mise en course tandis que la méthode GSU* réduit de près de 1341 minutes (68%) ce même temps par rapport à ne pas faire de famille. Ainsi, la proportion de temps entre le temps de mise en course et le temps d'opération total passe de 58% à 40.5% avec SDS et à 34.7% avec GSU*. De plus, le temps de passage moyen des commandes est réduit par le même coup de 497 minutes (27.1%) et 641 minutes (34.9%) respectivement.

Ordonnancement interfamilles

La figure 17 montre la différence qu'il y a entre les différentes règles d'ordonnancement interfamilles. Il est à noter que le temps de mise en course est le même pour chacune des règles interfamilles. Les règles RND et MS ne présentent pas de résultats significativement différents au niveau des en-cours. Par contre leurs en-cours sont plus faibles. De plus, la règle FSPT donne des en-cours les plus élevés de toutes les règles. Malgré tout, le temps de passage moyen de FSPT est inférieur de 179 minutes (13.6%) par rapport à Rnd. Tandis que la règle MS accroît le temps de passage moyen de plus de 136 minutes (10.3%) par rapport à RND.

Ordonnancement intrafamille

Cette section présente l'effet de l'ordonnancement intrafamille sur les quatre mesures de performance du système pour aucun groupement et lorsqu'il y a de la formation de familles.

Aucun groupement

Le tableau 14 présente les résultats des simulations du système de production lorsqu'il n'y a aucune formation de familles. Seulement l'ordonnancement intrafamille est considéré puisqu'il n'y a pas de familles et les résultats sont présentés à la figure 18. Les règles RND et RND/PCB produisent un niveau inférieur d'en-cours que les règles SPT et SPT/PCB. Le temps de mise en course est inférieur en regroupant les produits par type de PCB, soit RND/PCB ou SPT/PCB. La différence n'est pas significative entre les deux niveaux. L'imposition de garder les PCB de même types ensemble permet de sauver plus de 450 minutes, ou 19% de mises en course par semaine. La règle SPT/PCB donnera, par contre, un temps de passage moyen plus faible que les autres règles. De plus, la

règle SPT/PCB donne le temps de passage moyen le plus bas de toutes les règles étudiées.

Formation de familles

Le tableau 15 montre l'effet des différentes façons de grouper les produits sur les en-cours, le temps de passage, le temps de mise en course et le ratio du temps de mise en course sur le temps de production. La formation de familles avec la méthode SDS permet une réduction de 52% des temps de mise en course tandis que la méthode GSU* réduit de près de 68% ce même temps par rapport à ne pas faire de famille. Le temps de passage moyen des commandes est réduit par le même coup de 497 minutes (27%) et 641 minutes (35%) respectivement. Les règles Rnd et Rnd/PCB ne produisent aucune différence significative pour les en-cours. La même situation existe pour les règles SPT et SPT/PCB. Au niveau du temps de mise en course, il n'y a aucune distinction entre les règles Rnd/PCB et SPT/PCB. Quant au temps de passage, les règles SPT et Rnd/PCB ne donnent pas des résultats significativement différents.

Interactions

L'interaction entre la méthode de groupement et l'ordonnancement interfamilles permet de voir si l'effet entre les deux variables est significatif. (figure 19.) La méthode de groupement SDS produit un niveau d'en-cours plus élevé par rapport à GSU* pour les règles Rnd et MS tandis qu'il est plus faible avec les règles FSPT et FSPTS. Le temps de passage de GSU* est plus faible que SDS sauf pour Rnd.

L'interaction entre la méthode de groupement et l'ordonnancement intrafamille permet de voir si l'effet est plus prononcé dans un cas, comme le montre la figure 20. La méthode de groupement SDS est plus

sensible à l'utilisation de règles d'ordonnancement que GSU*. Les deux méthodes donnent de meilleurs résultats lorsque les PCB de même types sont gardés ensemble à l'intérieur de la famille.

Tableau 14
Résultats sans groupement de produits

Ordonnancement	En-Cours	T. Passage	T. Mise en course	Ratio
RND	617.8492993	2185.200889	2400.755768	0.60891288
SPT	915.5510686	1680.547302	2349.133541	0.60326722
RND/PCB	623.7424297	1911.401659	1902.810991	0.5530611
SPT/PCB	889.9215459	1567.500025	1895.498496	0.55317534

Tableau 15
Résultats des méthodes de groupement Expérience 2

Méthode de groupement	En cours	T Passage	T Mise en course	Ratio
Aucune	761.766	1836.162	2137.050	0.5796041
SDS	717.998	1339.063	1027.265	0.4052923
GSU*	735.334	1195.829	795.492	0.3465164

Tableau 16
Analyse de Duncan Expérience 2

Mesure de performance		En-cours		Temps de passage		Temps de mise en course	
Alpha = 0.05		DF = 128					
Carré moyen		724.5068		1274.211		226.643	
Variable	Niveau	Gr.	Moyenne	Gr.	Moyenne	Gr.	Moyenne
Méthode	SDS	B	717.998	A	1339.063	A	1027.265
	GSU*	A	735.334	B	1195.829	B	795.492
Interfamilles	Rnd	C	652.654	B	1312.393	A	911.377
	FSPT	A	813.131	D	1133.318	A	911.377
Intrrafamille	MS	C	645.170	A	1448.658	A	911.377
	FSPTS	B	795.709	C	1175.415	A	911.382
Intrrafamille	Rnd	B	708.771	A	1317.255	A	957.458
	SPT	A	744.569	B	1259.340	B	936.484
	Rnd/PCB	B	711.511	B	1263.386	C	876.282
	SPT/PCB	A	741.814	C	1229.804	C	875.289

Tableau 17

Analyse de variance du niveau d'en-cours Expérience 2

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	1053194.2127	33974.0069	46.89	0.0001
Error	128	92736.8715	724.5068		
Corrected Total	159	1145931.0841			
		R-Square	C.V.	Root MSE	WIP Mean
		0.919073	3.704131	26.916664	726.66607
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	12022.33703	12022.33703	16.59	0.0001
INTERF	3	974501.10162	324833.70054	448.35	0.0001
INTRAF	3	43995.05357	14665.01786	20.24	0.0001
METHODE*INTERF	3	19198.60318	6399.53439	8.83	0.0001
METHODE*INTRAF	3	2832.10730	944.03577	1.30	0.2764
INTERF*INTRAF	9	482.10012	53.56668	0.07	0.9999
METHOD*INTERF*INTRAF	9	162.90987	18.10110	0.02	1.0000

Tableau 18

Analyse de variance du temps de passage Expérience 2

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	3847460.1103	124111.6165	97.40	0.0001
Error	128	163099.0178	1274.2111		
Corrected Total	159	4010559.1281			
		R-Square	C.V.	Root MSE	TPAS Mean
		0.959333	2.816379	35.696093	1267.4462
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	820634.4384	820634.4384	644.03	0.0001
INTERF	3	2452723.6478	817574.5493	641.63	0.0001
INTRAF	3	159201.3014	53067.1005	41.65	0.0001
METHODE*INTERF	3	392929.7490	130976.5830	102.79	0.0001
METHODE*INTRAF	3	19497.5130	6499.1710	5.10	0.0023
INTERF*INTRAF	9	1780.4303	197.8256	0.16	0.9977
METHOD*INTERF*INTRAF	9	693.0304	77.0034	0.06	1.0000

Tableau 19

Analyse de variance du temps de mise en course Expérience 2

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	2381311.6326	76816.5043	338.93	0.0001
Error	128	29010.3050	226.6430		
Corrected Total	159	2410321.9375			
	R-Square	C.V.	Root MSE		SETUPMean
	0.987964	1.651857	15.054667		911.37832
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	2148740.3867	2148740.3867	9480.73	0.0001
INTERF	3	0.0006	0.0002	0.00	1.0000
INTRAF	3	211512.4583	70504.1528	311.08	0.0001
METHODE*INTERF	3	0.0006	0.0002	0.00	1.0000
METHODE*INTRAF	3	21058.7832	7019.5944	30.97	0.0001
INTERF*INTRAF	9	0.0017	0.0002	0.00	1.0000
METHOD*INTERF*INTRAF	9	0.0017	0.0002	0.00	1.0000

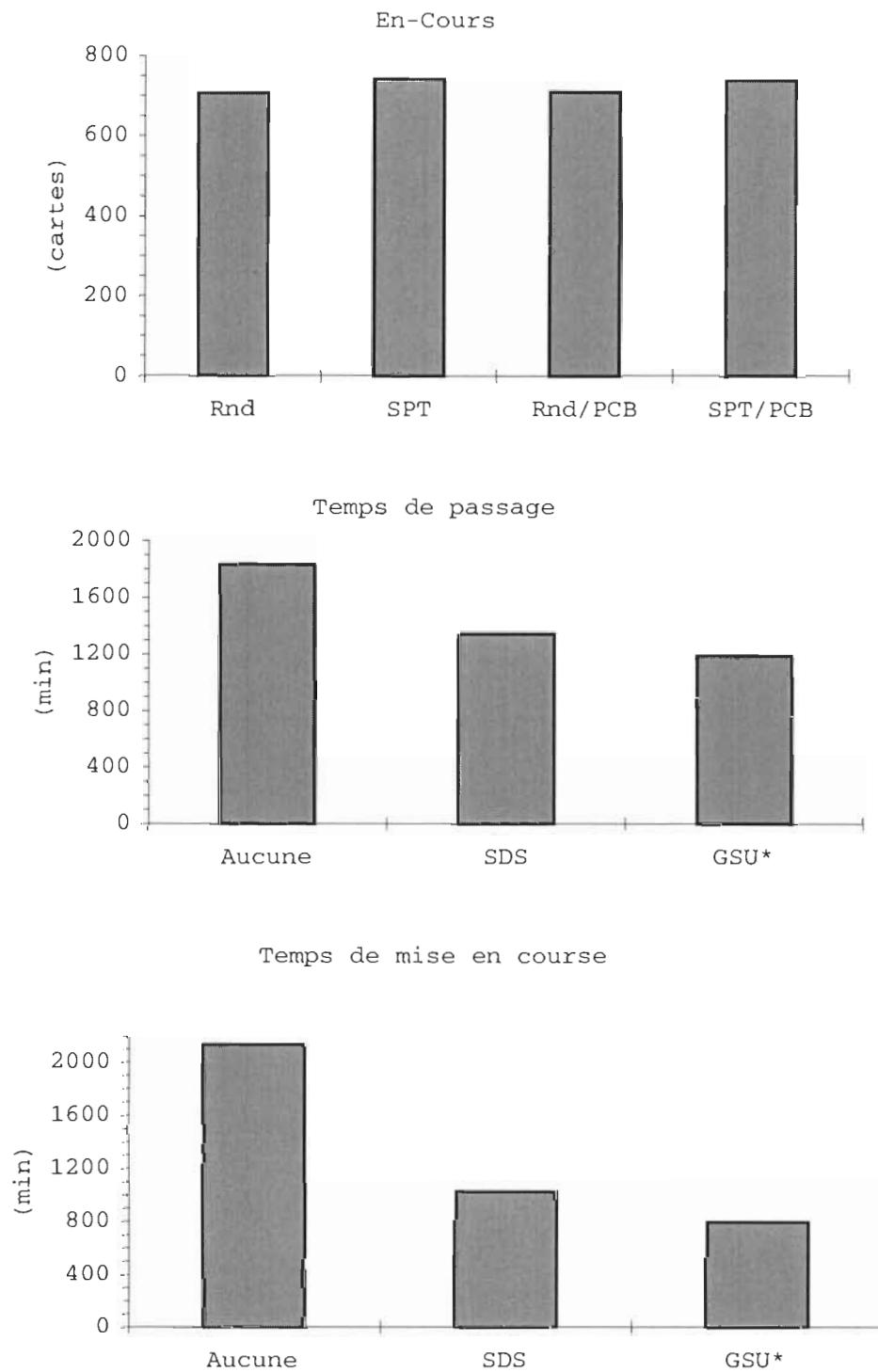


Figure 16. Niveau des méthodes de groupement Expérience 2

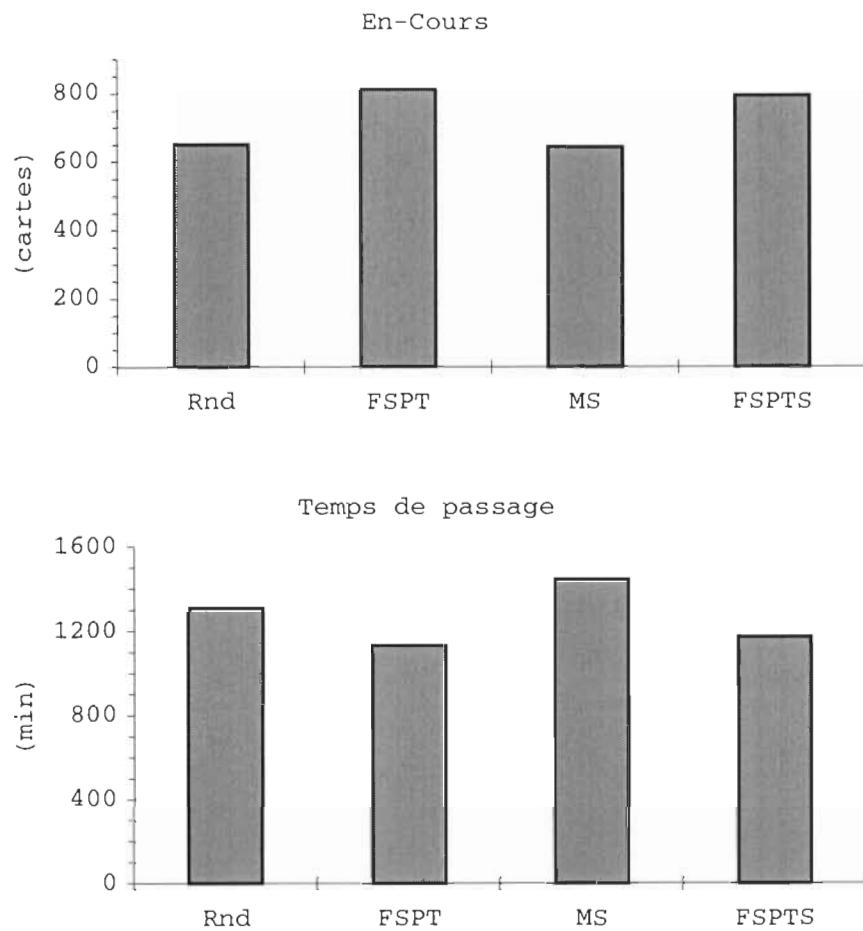


Figure 17. Niveau de l'ordonnancement interfamilles Expérience 2

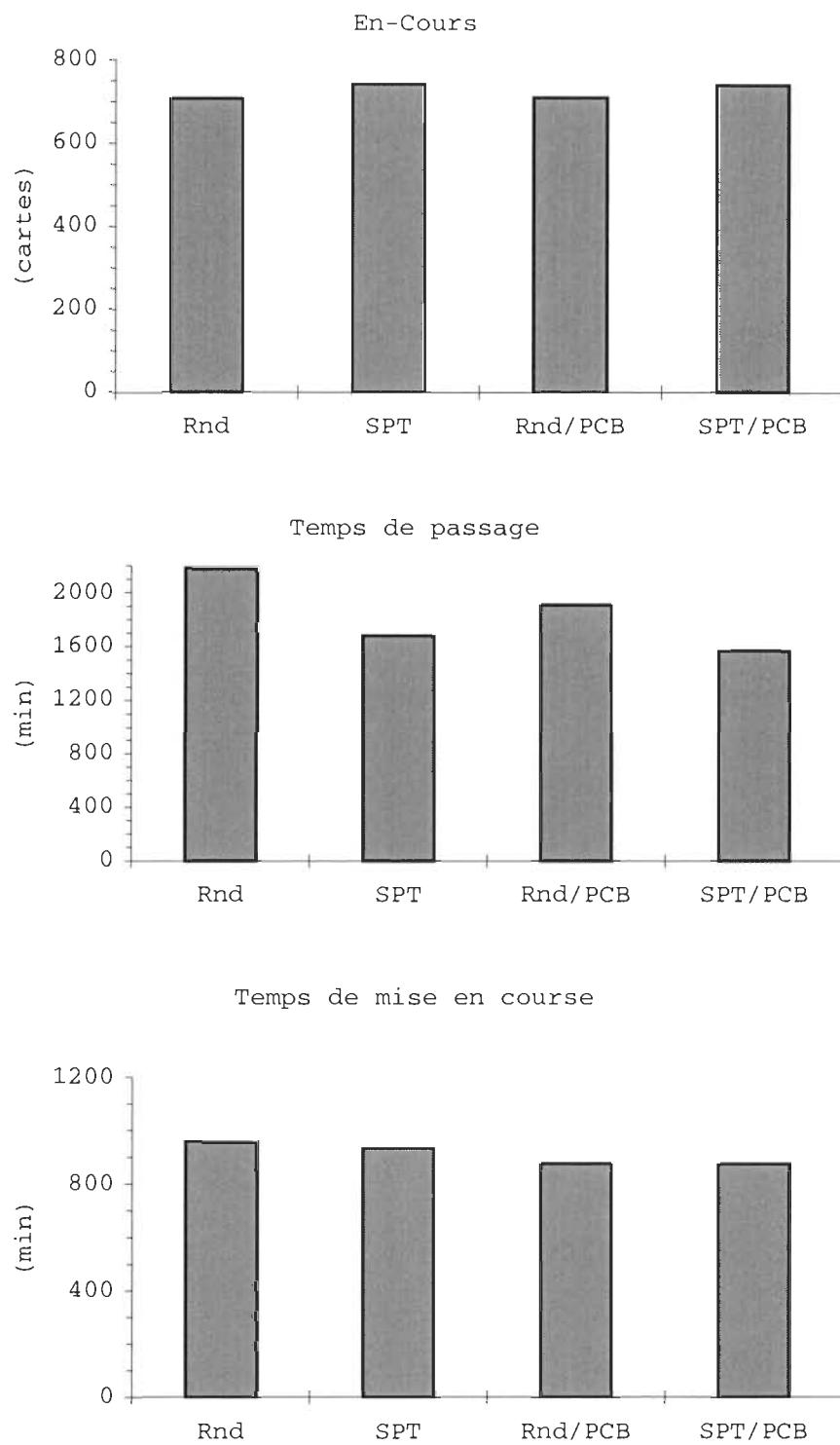


Figure 18. Intrafamille sans formation de famille

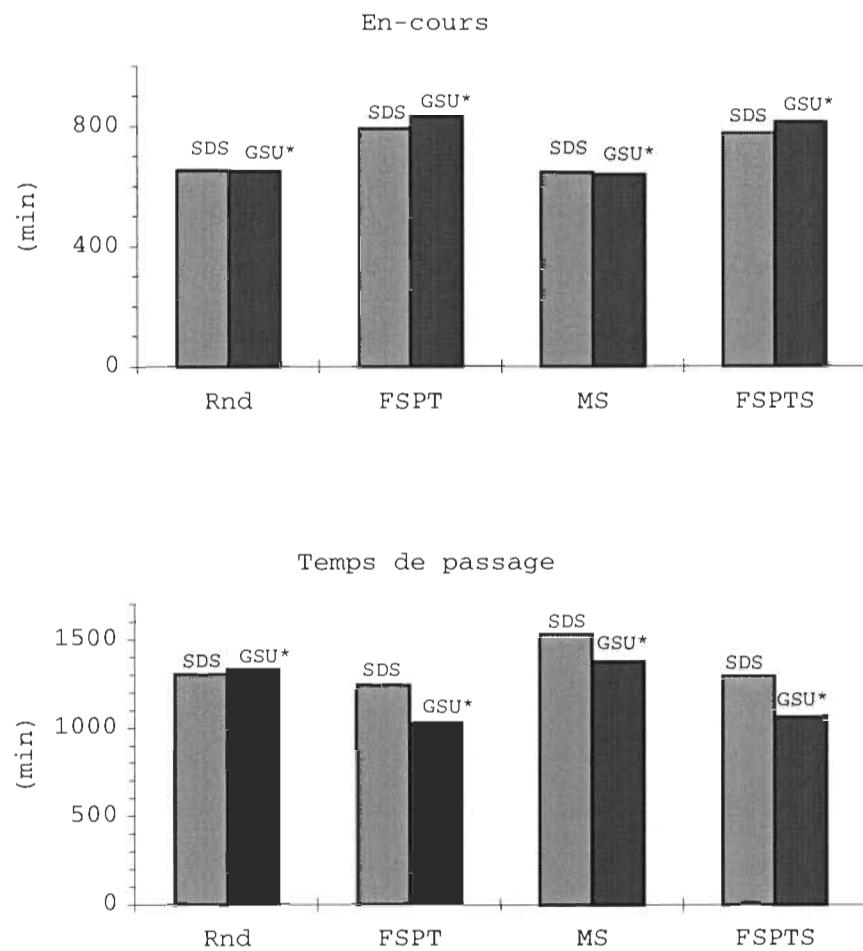


Figure 19. Interaction méthodes de groupement et ordonnancement
interfamilles Expérience 2

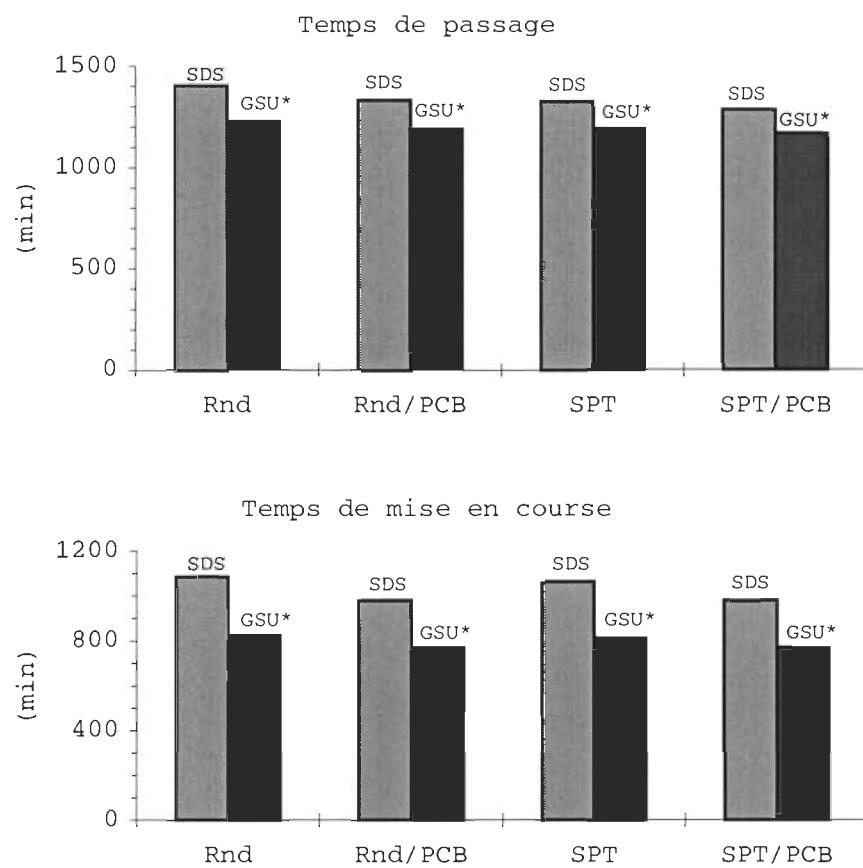


Figure 20. Interaction méthodes de groupement et ordonnancement
intrafamille Expérience 2

CHAPITRE VI

CONCLUSION

Les modèles de simulation étudiés sont significativement représentatifs du système à l'étude. En effet, les résultats obtenus sont semblables au système de Matrox à tous les niveaux. Chacun des modèles étudiés pour les mesures de performance retenues a un R-carré de plus de 90% dans les deux expériences.

Discussion

Expérience 1

Les deux méthodes de groupement de produits choisies dans cette étude produisent des résultats différents selon les conditions du système comme le montre le tableau 20. La méthode GSU* en général est moins sensible que SDS au changement de niveau des autres variables. La plus grande fluctuation de GSU* s'effectue au niveau de l'en-cours. GSU* combiné avec l'utilisation des cassettes doubles diminuera les en-cours tandis que SDS les augmentera. Ceci semble attribuable à la quantité de produits qui peuvent être regroupés par famille. Dans le cas de cassettes simples, il y a environ 6 à 7 produits par famille avec SDS tandis que GSU* n'a que 2 à 3 produits. Le nombre de produits par famille avec SDS est de 10 à 12 et avec GSU* il est de 3 à 4 avec l'utilisation des cassettes doubles. Ainsi avec la méthode GSU*, le facteur de similarité imposé empêche des produits d'être inclus parce que les pièces communes ont déjà été chargées sur la machine. Le facteur de similarité à un

niveau de 10%, par rapport à aucun, ne fait pas ou très peu varier la composition des familles selon la méthode SDS. Aussi, plus la quantité de pièces communes sur la machine est grande moins il y a de chances d'avoir deux produits avec au moins 10% de similarité. Finalement, une nouvelle expérience sans imposer de niveau de similarité devrait être effectuée pour déterminer si l'utilisation de cassettes doubles avec la méthode GSU* pourrait donner de bons résultats.

La réduction de pièces donne des résultats décevants. Ceci est dû au manque de similarité entre les produits et aussi au grand nombre de produits différents à assembler à chaque semaine. L'effet réel de la réduction de pièces sur une semaine donnée est d'environ 1% bien que la liste principale de pièces puisse être réduite de près de 20%.

Tableau 20
Effet des variables sur la méthode de groupement

	En-cours		T. de passage		Mise en course	
	GSU*	SDS	GSU*	SDS	GSU*	SDS
Cassette double vs simple	↓	↑	↓	↓↓	↓	↓↓
Rho 0.3 vs 0.1	↑	↓	↑	↑↑	↑	↑↑
Rnd vs FSPT (Inter)	↑	↑	↓	↑	---	---
Rnd/PCB vs SPT/PCB (Intra)	↑	↑↑	↓	↓	≈	↑

L'utilisation de règles d'ordonnancement, peu importe la méthode de groupement choisie, peut améliorer le temps de passage des commandes dans le système sans toutefois trop augmenter le niveau des en-cours. La règle d'ordonnancement interfamilles FSPT permet de réduire de beaucoup le temps de passage moyen des commandes par rapport à Rnd. Il faut, par contre, accepter une augmentation du niveau des en-cours. La règle

d'ordonnancement intrafamille SPT/PCB donne un meilleur temps de passage que Rnd/PCB. Toutefois, la règle Rnd/PCB donne moins d'en-cours et une temps de mise en course d'à peine 10 minutes plus faible. L'ordonnancement interfamilles affecte plus la performance du système par rapport à l'ordonnancement intrafamille. La règle d'ordonnancement interfamilles FSPT et la méthode GSU* réduisent le temps de passage par rapport à Rnd. Une augmentation du temps de passage est observée lorsque la méthode SDS est utilisée avec la règle FSPT. La deuxième partie a été faite pour répondre entre autres à cette question.

Expérience 2

L'expérience 2 fait suite aux interrogations laissées par l'expérience 1. Une première est à savoir s'il est favorable de garder les PCB groupés au niveau de l'ordonnancement intrafamille. Quel est l'effet sur le système dû à la formation de familles? Est-ce qu'il y a d'autres règles d'ordonnancement interfamilles pouvant améliorer le temps de passage des commandes?

La formation de familles permet de réduire grandement les temps de mise en course. La méthode GSU* est supérieure à SDS à ce niveau. De plus, le temps de passage de la première sera inférieur. Par contre, SDS aura un niveau d'en-cours plus faible. L'augmentation du niveau d'en-cours de GSU* par rapport à SDS est faible parce que les produits n'ont pas à revenir en arrière pour un deuxième assemblage comme GSU demande.

L'impact de l'ordonnancement des produits lorsqu'il n'y a pas de formation de familles a aussi été étudié. La première constatation faite est que le temps de passage diminue avec n'importe quelle règle par rapport à ordonner aléatoirement. La variation du niveau des en-cours

moyen entre les différentes règles n'est pas bien grande. Finalement, le temps de mise en course sera plus faible si les PCB de même types sont regroupés ensemble.

L'ordonnancement interfamilles fait varier le temps de passage moyen de plusieurs heures, tout comme dans l'expérience 1. La règle MS donne le temps de passage le plus long de toutes les règles à l'étude. Par contre, les règles MS et Rnd donnent le niveau d'en-cours le plus bas. La règle FSPTS amène un temps de passage un peu plus long que FSPT. De plus, les résultats obtenus avec la règle MS ne sont pas très différents de Frazier (1996). Ainsi, il est préférable de ne pas tenir compte que du temps de mise en course des machines dans le choix de l'ordre des familles bien que les en-cours seront plus élevés. L'interaction observée dans cette partie ne peut répondre à la question soulevée précédemment puisque le facteur de similarité est toujours fixé à 10% pour toutes les expériences.

L'ordonnancement intrafamille donnera une réduction du temps de passage faible pour les deux méthodes de groupement. La règle SPT/PCB donnera le meilleur résultat au niveau du temps de passage et du temps de mise en course pour les deux méthodes. Elle entraîne, par contre, une augmentation du niveau des en-cours.

Conclusion

La formation de familles est favorable pour l'entreprise avec une grande variété de produits à assembler à chaque semaine. Elle permet de réduire le temps de mise en course hebdomadaire d'environ 50%. Ces heures gagnées pourront être consacrées à la production.

La méthode GSU* donne d'excellents résultats. De plus, les points reprochés à la méthode GSU ne sont pas aussi importants avec GSU*. Il y a en effet une augmentation des en-cours mais elle est faible. La méthode GSU* proposée est beaucoup plus simple pour la gestion des mises en course et des programmes des machines de placement que GSU.

Les règles d'ordonnancement interfamilles réduisent plus le temps de passage que l'ordonnancement intrafamille. Il peut être très difficile d'ordonnancer les tâches et familles dans un système soumis à des priorités ou contraintes hors de contrôle. Par contre, cette étude ne touche que la portion SMT du système de production. La répercussion sur les autres départements de recevoir les commandes plus tôt n'est pas connue.

L'expérience a permis de constater qu'une diminution du temps de mise en course entraîne directement une diminution du temps de passage moyen des commandes.

Recommandations

Une suite logique à cette recherche serait de poursuivre l'étude pour voir l'impact sur le système de production en entier des règles d'ordonnancement interfamilles et intrafamille. De plus, l'algorithme utilisé pour la sélection des produits à inclure dans les familles devrait être revu.

Avant d'utiliser la méthode GSU*, une étude de similarité des produits devrait être faite pour déterminer le nombre optimal d'espaces

cassettes à allouer pour les pièces les plus utilisées dans la semaine. Aussi, cette étude devrait revoir le niveau du facteur de similarité de façon à permettre une meilleure formation de familles.

BIBLIOGRAPHIE

High-tech companies decrease cycle time. IIE Solutions, Mai 1996, p. 8.

Baker, K. R., Powell, S. G., & Pyke, D. F. The performance of push and pull systems : a corrected analysis. International Journal of Production Research, Vol. 28, No. 9, 1990, pp. 1731-1736.

Berkley, B. J. Simulation Tests of FCFS and SPT Sequencing in Kanban Systems. Decision Sciences, Vol. 24, 1993, pp. 218-227.

Berkley, B. J., & Kiran, A. S. A Simulation Study of Sequencing Rules in a Kanban-Controlled Flow Shop. Decision Sciences, Vol. 22, 1991, pp. 559-582.

Bhaskar, G., & Narendran, T. T. Grouping PCBs for set-up reduction: a maximum spanning tree approach. International Journal of Production Research, Vol. 34, No. 3, 1996, pp. 621-632.

Brooks, B. M. Concurrent Engineering - How do I know when I'm doing it?. Proceedings of the International Conference on Concurrent Engineering and Electronic Design Automation (CEEDA 194), 1994, pp. 23-28.

Broughton, T. Simultaneous Engineering in Aero Gas Turbine Design and Manufacture. Proceedings of the 1st International Conference on Simultaneous Engineering, 1990.

Carmon, T. F., Maimon, O. Z., & Dar-El, E. M. Group set-up for printed circuit board assembly. International Journal of Production Research, Vol. 27, No. 10, 1989, pp. 1795-1810.

Carter, D. E. Concurrent Engineering. Proceedings of the International Conference on Concurrent Engineering and Electronic Design Automation (CEEDA 194), 1994, pp. 5-7.

Co, H. C., & Jacobson, S. H. The kanban assignment problem in serial Just-In-Time production systems. IIE Transactions, Vol. 26, No.2, Mars 1994, pp.76-85.

Cyr, B. Effets de différents facteurs sur la flexibilité d'une ligne de montage en surface : une étude de simulation. Mémoire non-publié, Août 1997.

Eversheim, W. Trends and Experience in Applying Simultaneous Engineering in Germany. Proceedings of the 1st International Conference on Simultaneous Engineering, 1990.

Fisher, D. The Just-In-Time Self Test. Irwin Professional Publishing, 1995.

Ford, H. Today and Tommorow. Reproduction de la bibliographie originale 1926, Production Press, 1988.

Frazier, G. V. An evaluation of group scheduling heuristics in a flow-line manufacturing cell. International Journal of Production Research, Vol. 34, No. 4, 1996, pp. 959-979.

Hashiba, S., & Chang, T. C. PCB Assembly Setup Reduction Using Group Technology. Computers and Industrial Engineering, No. 21, 1991, pp. 453-457.

Hegstad, M. A simple, low-risk, approach to JIT. Production Planning & Control, Vol. 1, No. 1, 1990, pp. 53-60.

Kusiak, A. Concurrent Engineering: Automations, Tools, and Techniques, Edited by Andrew Kusiak., New York: John Wiley & Sons, 1992.

Lambert, S. Caractérisation de l'entreprise Les Systèmes Électroniques Matrox Ltée vis-à-vis la philosophie de production dite Juste-À-Temps., Rapport confidentiel, Août 1996.

Lee, L. C. Parametric appraisal of the JIT system. International Journal of Production Research, Vol. 25, 1987, pp. 1415-1429.

Maimon, O., Carmon, T. F., & Dar-El, E. M. Set-up saving schemes for printed circuit board assembly. European Journal of Operations Research, 1993, No. 70, pp. 177-190.

Maimon, O., & Shtub, A. Grouping methods for printed circuit board assembly. International Journal of Production Research, Vol. 29, No. 7, 1991, pp. 1379-1390.

Monden, Y. Toyota production system : An integrated approach to Just-In-Time (2nd ed.). IIE, 1993.

Panwalker, S. S., & Iskander, W. A Survey of Scheduling Rules. Operations Research, Vol. 25, No. 1, 1977, pp. 45-61.

Rees, L. P., Huang, P. Y., & Taylor III, B. W. A comparative analysis of an MRP lot-for-lot system and a Kanban system for a multistage production operation. International Journal of Production Research, Vol. 27, No. 8, 1989, pp. 1427-1443.

Shtub, A., & Maimon, O. Role of similarity measures in PCB grouping procedures. International Journal of Production Research, Vol. 30, No. 5, 1992, pp. 973-983.

Spearman, M. L., & Zazanis, M. A. Push and Pull Production Systems: Issues and Comparisons. Operations Research, Vol. 40, No. 3, 1992, pp. 521-532.

Wantuck, K. Just-in-time for America, KWA Media, 1989.

Wedderburn, H. A just-in-time perspective. Industrial Management, Juillet 1985, pp. 4-8.

Zygmont, J. Just-In-Time controls more than inventory. Managing Automation, Octobre 1989, pp. 30-32.

ANNEXE A

DESCRIPTION DES PROGRAMMES

Plusieurs programmes DOS sont nécessaires pour accomplir la simulation. Le premier programme nécessaire est gentxt.exe. Ce programme existait déjà dans l'entreprise et n'a pas été modifié. Il sert à lire le fichier de commandes généré par la simulation test.res. Pour chacun des produits contenus, il accède la base de données des produits pour aller chercher la liste de pièces de chacun des produits (fichier.smt) et génère les fichiers liste.txt et dum.txt.

Les programmes setfam10.exe et setfam30.exe sont identiques au niveau du fonctionnement sauf pour la valeur du facteur de similarité rho. L'un est à 10% et l'autre est à 30%. Les programmes, à partir des fichiers liste.txt, dum.txt, feeder.txt génèrent le fichier setup.txt. La figure 21 montre le fonctionnement du programme setfam10.exe.

Les modifications suivantes ont été faites à la fonction choisi_prochain() du programme setfam.exe de Matrox pour tenir compte du facteur de similarité. La condition sur rho s'ajoute aux autres conditions; places restantes, espace disponible et produit non choisi.

```
// Choisi le plus petit sub de la liste
if ((nom_sub[i].pl > 0) && (nom_sub[i].esp < j)
    && (nom_sub[i].famille == 0)
    && ((100 * (nom_sub[i].pr-nom_sub[i].pl)/nom_sub[i].pr)>10))
{
    j = nom_sub[i].esp;
    l = j;
    k = i + 1;
    sub_choisie = nom_sub[i].nom;
```

```
        }
    }
    if ((nom_sub[i].pr==0) || (nom_sub[i].pl==0)) rho = 100;
    else rho = (100 * (nom_sub[i].pr-nom_sub[i].pl)/nom_sub[i].pr);
    printf("%d - Nom: %s Piece restante : %d (esp:%d) - rho: %d\n",i,
nom_sub[i].nom, nom_sub[i].pl, nom_sub[i].esp, rho);
}
```

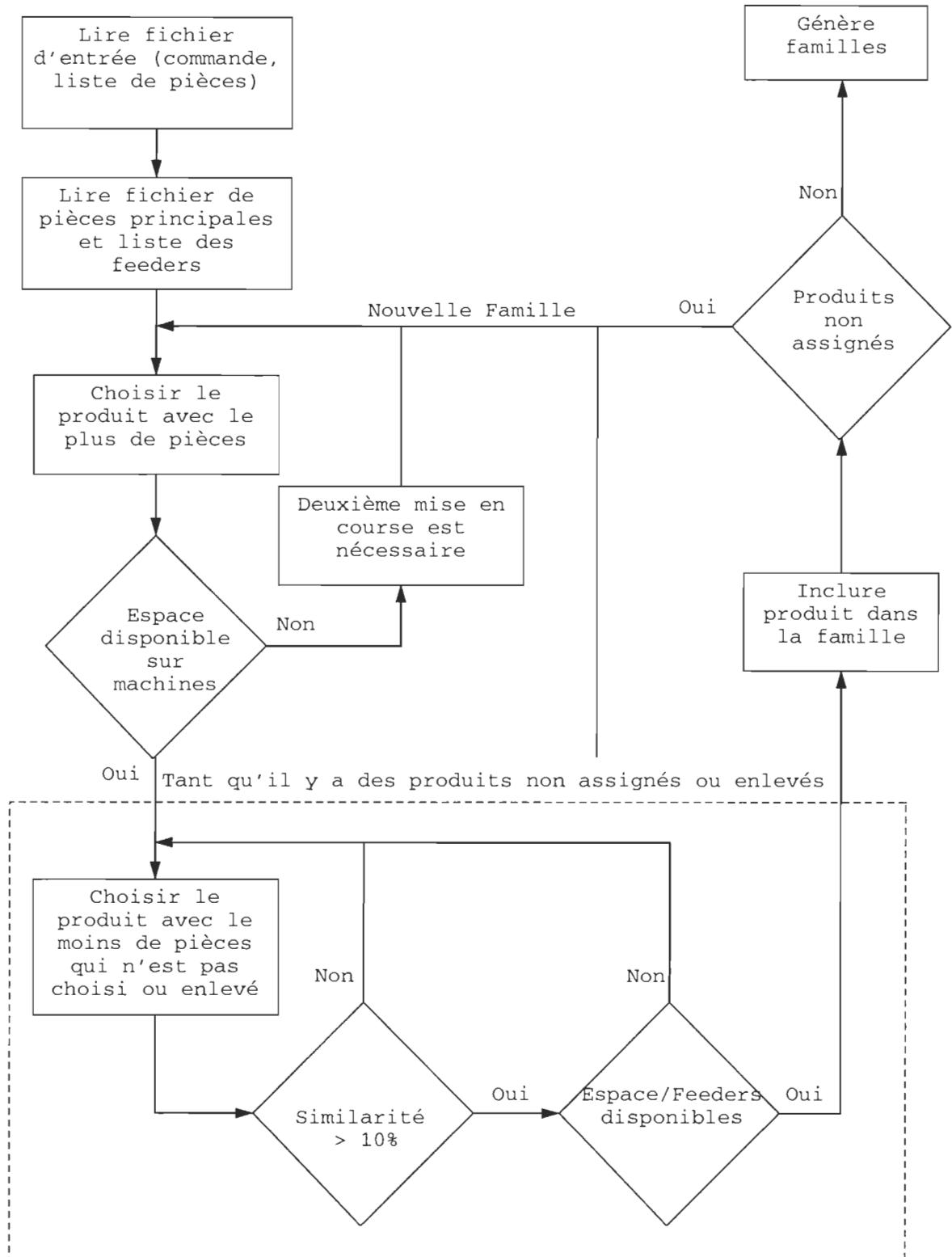


Figure 21. Programme Setfam10

Le programme pieces.exe sert pour la réduction de pièces. Il lit le fichier liste.txt et dum.txt et les ajuste pour tenir compte des pièces qui peuvent être standardisées. De nouveaux fichiers liste.txt et dum.txt sont créés pour tenir des changements.

```
-----
// Reduction du nombre de pieces par produit
//-----  

// Par: Serge Lambert
// Date: 20 janvier 1997
//-----  

# include <stdio.h>
# include <string.h>
# include <stdlib.h>
# include <malloc.h>
# define MAX_SUB 100
# define MAX_PIECES 3120  

// Variables pour liste.txt
typedef struct
{
    char smt[13];
    char nom[16];
    unsigned char no_sub;
    int qte;
} Sub;  

// Variables pour dum.txt
typedef struct
{
    char nom[16];
    unsigned char qte;
    unsigned char no_pcb;
    unsigned char flag;
} Dum;  

FILE *finl, *fin2, *fout, *fout1;  

int nombre_sub, nombre_piece;  

Sub produit[MAX_SUB];
Dum liste[MAX_PIECES];  

void main(void)
{
    int i, j, k, l, m;
    char tampon[4];  

    // Tableau avec la liste des produits a assembler
    if ((finl = fopen("liste.txt", "r")) == NULL)
    {
```

```

        printf("Je ne trouve pas le fichier liste.txt\n");
        exit(-1);
    }
    i = 0;
    while (fscanf(finl,"%d %d %d %d %s %s %s", &j, &l, &m, &k, produit[i].smt,
tampon, produit[i].nom) == 7)
    {
        produit[i].no_sub = 1;
        produit[i].qte = m;
        i++;
    }
    nombre_sub = i;
    fclose(finl);

    // Tableau avec la liste des pieces par sub
    if ((fin2 = fopen("dum.txt", "r")) == NULL)
    {
        printf("Je ne trouve pas le fichier util.txt\n");
        exit(-1);
    }
    i = 0;
    while (fscanf(fin2,"%d %s %d", &l, liste[i].nom, &m) == 3)
    {
        liste[i].no_pcb = 1;
        liste[i].qte = m;
        i++;
    }
    nombre_piece = i;
    fclose(fin2);
    for (i=0; i<nombre_piece; i++)
    {
        if (liste[i].flag == 0)
        {
            // Standardisation des condensateur au niveau de la tolerance
            if(strncmp(liste[i].nom, "CC", 2)==0)
            {
                for (j=i; j<nombre_piece; j++)
                {
                    if(strcmp(liste[i].nom, liste[j].nom, 6)==0)
                    {
                        liste[i].flag = 1;
                        strcpy(liste[j].nom, liste[i].nom);
                        liste[j].flag = 1;
                    }
                }
            }
            // Standardisation des circuits integres PAL au niveau de la
vitesse
            if(strncmp(liste[i].nom, "IP", 2)==0)
            {
                for (j=i; j<nombre_piece; j++)
                {
                    if(strcmp(liste[i].nom, liste[j].nom, 8)==0)
                    {
                        liste[i].flag = 1;
                        strcpy(liste[j].nom, liste[i].nom);
                        liste[j].flag = 1;
                    }
                }
            }
            // Standardisation circuits integres TTL au niveau de la vitesse
            if(strncmp(liste[i].nom, "IT", 2)==0)
            {
                for (j=i; j<nombre_piece; j++)
            }
        }
    }
}

```

```

        {
            if(strncmp(liste[i].nom, liste[j].nom, 8)==0)
            {
                liste[i].flag = 1;
                strcpy(liste[j].nom, liste[i].nom);
                liste[j].flag = 1;
            }
        }
    }
    // Standardisation des EPROM au niveau de la vitesse
    if(strcmp(liste[i].nom, "MU", 2)==0)
    {
        for (j=i; j<nombre_piece; j++)
        {
            if(strcmp(liste[i].nom, liste[j].nom, 8)==0)
            {
                liste[i].flag = 1;
                strcpy(liste[j].nom, liste[i].nom);
                liste[j].flag = 1;
            }
        }
    }
    // Standardisation des resistances S au niveau de la tolerance
    if(strcmp(liste[i].nom, "RS", 2)==0)
    {
        for (j=i; j<nombre_piece; j++)
        {
            if(strcmp(liste[i].nom, liste[j].nom, 7)==0)
            {
                liste[i].flag = 1;
                strcpy(liste[j].nom, liste[i].nom);
                liste[j].flag = 1;
            }
        }
    }
}

// Ecrit le fichier reduit dum
if ((fout1 = fopen("dumred.txt", "w")) == NULL)
{
    printf("Je ne peux pas ouvrir le fichier dumred.txt\n");
    exit(-1);
}
k = 0;
for (i=0; i<nombre_piece; i++)
{
    if (liste[i].flag == 1)
    {
        for (j=i+1; j<nombre_piece; j++)
        {
            m = 0;
            if (((strcmp(liste[i].nom,
liste[j].nom)==0)&&(liste[i].no_pcb == liste[j].no_pcb)))
            {
                liste[i].qte += liste[j].qte;
                liste[j].qte = 0;
                m = liste[j].no_pcb;
                produit[m-1].qte--;
            }
        }
        if(liste[i].qte>0)
        {

```

```
        fprintf(fout1, "%d      %-20s      %d\n",      liste[i].no_pcb,
liste[i].nom, liste[i].qte);
    }
    else
    {
        fprintf(fout1, "%d  %-20s  %d\n",  liste[i].no_pcb,  liste[i].nom,
liste[i].qte);
    }
    fclose(fout1);

// Ecrit le tableau de la liste des sub
if ((fout = fopen("listered.txt", "w")) == NULL)
{
    printf("Je ne trouve pas le fichier listered.txt\n");
    exit(-1);
}

for (i=0; i<nombre_sub; i++)
{
    fprintf(fout, "0  %-3d  %-3d  0  %s  MES  %s\n",  produit[i].no_sub,
produit[i].qte, produit[i].smt,
            produit[i].nom);
}
fclose(fout);
}
```

Le programme util.exe est utilisé pour déterminer les pièces les plus utilisées pour la semaine de production. Il compte le nombre de produits qui utilisent une pièce. Les pièces sont mises en ordre en fonction de leur utilisation et écrit dans le fichier util.txt.

```

// Ver Date      Nom          Description de la Modification      Backup
// 1  27/01/97  S.Lambert  Creation du fichier util.txt          util.c

#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char tmp[1400][16];
char prod[1400][16];
unsigned char qte[1400];
unsigned char pcb[1400];
int freq[1400];
int total[1400];

void main(void)
{
    FILE *util,*idiot;
    int i,j,k,l,flag;
    int vpcb, vtotal;
    int pcb_temp, total_temp;
    char tempo[17],qte_temp[2],freq_temp[4];
    char vtempo[17];
    flag = 0;
    i = 0;
    if ((util = fopen("util.txt", "w")) == NULL)
    {
        printf("Cannot open output file %s\n", "util.txt");
        exit(1);
    }
    if ((idiot = fopen("dum.txt", "r")) == NULL)
    {
        printf("Cannot open input file %s\n", "dum.txt");
        exit(1);
    }
    while (!feof (idiot))
    {
        fscanf(idiot,"%s %s %s", qte_temp, tempo, freq_temp);
        if (i>0)
        {
            for (j=0; j<i; j++)
            {
                if(strcmp(tempo, tmp[j])==0)
                {
                    flag = 1;
                    break;
                }
            }
        }
        if (flag==1)
        {

```

```

        qte[j] = qte[j]+ 1;
        freq[j] = freq[j] + atoi(freq_tmp);
        flag = 0;
    }
    else
    {
        strcpy(tmp[i], tempo);
        if(strcmp(tempo, "")!=0)
        {
            qte[i] = 1;
            freq[i] = atoi(freq_tmp);
            i++;
        }
    }
    strcpy(tempo, "");
}
for (j=0; j<i; j++)
{
    strcpy(vtempo, tmp[j]);
    vpcb = qte[j];
    vtotal = freq[j];
    if (j==0)
    {
        strcpy(prod[j], vtempo);
        pcb[j] = vpcb;
        total[j] = vtotal;
    }
    else if (j>0)
    {
        k = j;
        for (l = 0; l < k; l++)
        {
            j--;
            if (pcb[j] > vpcb)
            {
                strcpy(tempo, prod[j]);
                pcb_temp = pcb[j];
                total_temp = total[j];
                j++;
                strcpy(prod[j], tempo);
                pcb[j] = pcb_temp;
                total[j] = total_temp;
                j--;
                if (j==0)
                {
                    strcpy(prod[j], vtempo);
                    pcb[j] = vpcb;
                    total[j] = vtotal;
                }
            }
            else
            {
                j++;
                strcpy(prod[j], vtempo);
                pcb[j] = vpcb;
                total[j] = vtotal;
            }
        }
        j = k;
    }
}
for (j=0; j<i; j++)
{
    fprintf(util,"%-17s  %-3d  %-5d\n", prod[j], pcb[j], total[j]);
}

```

```
    }
    fclose(util);
    fclose(idiot);
}
```

Le programme gsu.exe reprend le fichier util.txt pour générer le fichier setupgsu.txt. Ce fichier contient les pièces devant être chargées sur les espaces dédiés pour les pièces communes pour la MPA et la MVII. De plus, le programme ajuste les fichiers liste.txt et dum.txt pour tenir compte des pièces de la partie dédiée.

```
-----
// PROGRAMME DES FAMILLES DYNAMIQUES
//-----  

// Ver Date      Nom          Description de la Modification      Backup
// 1  11/01/97  S.Lambert  Adaptation pour GSU          Gsu.c
//-----  

# include <stdio.h>
# include <string.h>
# include <stdlib.h>
# include <malloc.h>

# include "setup.h";

int nb_feeder;
int tray;
int place_piece = ESPACE_TOTAL;
int global_var;
int global_total, espace_mv2=0, espace_mpa=0, mga;

FILE *erreur;
FILE *gsu, *feed, *lis, *dum;
int nombre_piece;
int nombre_sub;
int nombre_produit;
char piece_temp[16];

// Contient les pieces avec les subs qui les utilisent
Piece liste[MAX_PIECES];
Feeder feeder_mv2[TYPE_FEEDER], feeder_mpa[TYPE_FEEDER];
Sub produit[MAX_SUB];

//-----  

// Lire le nombre de feeder de la MV2 et MPA
//-----  

void lire_feeder()
{
    FILE *fi;
    int i, qty_mpa, qty_mv2, espace ;
    if ((fi = fopen("feeder.txt", "r")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier feeder.txt\n");
        fprintf(erreur,"%s","Je ne trouve pas le fichier feeder.txt\n");
        exit(-1);
    }
}
```

```

i = 0;
while (fscanf(fi, "%s %d %d %d", feeder_mpa[i].nom, &qty_mpa, &qty_mv2,
             &espace) == 4)
{
    strcpy(feeder_mv2[i].nom, feeder_mpa[i].nom);
    feeder_mv2[i].qty = feeder_mv2[i].init = qty_mv2;
    feeder_mpa[i].qty = feeder_mpa[i].init = qty_mpa;
    feeder_mv2[i].espace = feeder_mpa[i].espace = espace;
    i++;
}
nb_feeder=i;
}

//-----
//          Lire les noms de pieces
//-----
void lire_utiliser()
{
    FILE *fu;
    int i;
    if ((fu = fopen("util.txt", "r")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier util.txt\n");
        fprintf(erreur,"%s","Je ne trouve pas le fichier util.txt\n");
        exit(-1);
    }
    nombre_piece = i = 0;
    while (fscanf(fu, "%s %d %d", liste[nombre_piece].nom,
                  liste[nombre_piece].no_pcb, liste[nombre_piece].qte) == 3)
    {
        nombre_piece++;
        i++;
        if (i >= MAX_PIECES)
        {
            fprintf(stderr, "Trop de pieces...\n");
            fprintf(erreur,"%s", "Trop de pieces...\n");
            exit(-1);
        }
    }
}

//-----
//          Lire les types de feeders pour les pieces
//-----
void lire_piece()
{
    FILE *fi;
    int i,j,k;
    char ptemp[16], feeder_temp[6];
    if ((fi = fopen("piece.txt", "r")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier piece.txt\n");
        fprintf(erreur,"%s","Je ne trouve pas le fichier piece.txt\n");
        exit(-1);
    }
    i = 0;
    while (fscanf(fi, "%s %s", ptemp, feeder_temp) == 2)
    {

```

```

        for (j=0; j <nombre_piece; j++)
        {
            if (strcmp(liste[j].nom, ptemp) == 0 )
            {
                strcpy(liste[j].type_feeder, feeder_temp);
                for (k=0; k<TYPE_FEEDER; k++)
                {
                    if (strcmp(feeder_mv2[k].nom, feeder_temp)==0)
                    {
                        liste[j].espace = feeder_mv2[k].espace;
                    }
                }
                liste[j].flag = 0;
                i++;
                break;
            }
        }
    }

//-----
//          Lire les subs
//-----
void lire_liste()
{
    FILE *fl;
    int i,j,k,l,m;
    char tampon[4];

    if ((fl = fopen("liste.txt", "r")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier liste.txt\n");
        fprintf(erreur,"%s","Je ne trouve pas le fichier liste.txt\n");
        exit(-1);
    }
    i = 0;
    while (fscanf(fl,"%d %d %d %d %s %s %s", &j, &l, &m, &k, produit[i].smt,
                  tampon, produit[i].nom) == 7)
    {
        produit[i].no_sub = 1;
        produit[i].qte = m;
        i++;
    }
    nombre_sub = i;
}

//-----
//          Mise a jour des liste de piece
//-----
void mise_a_jour()
{
    FILE *fd;
    int j,k, nosub;
    char ptemp[16];

    if ((fd = fopen("dum.txt", "r+")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier dum.txt\n");
        fprintf(erreur,"%s","Je ne trouve pas le fichier dum.txt\n");
    }
}

```

```

        exit(-1);
    }

    while(fscanf(fd, "%d %s %d", &nosub, ptemp, &k) == 3)
    {
        if (strcmp(piece_temp, ptemp)==0)
        {
            for (j=0; j<nombre_sub; j++)
            {
                if (nosub == produit[j].no_sub) produit[j].qte--;
            }
        }
        strcpy(ptemp, "");
    }
    fclose(fd);
}

//-----
// Choisi une piece
//-----
void choisi_piece()
{
    int i, k, total, qty_mv2, qty_mpa, espace, libre_mv2, libre_mpa;

    total = 0;
    for (i = nombre_piece; i >= 0; i--)
    {
        if ((liste[i].flag)==0)
        {
            global_var++;
            total += liste[i].espace;
            for (k = 0; k < TYPE_FEEDER; k++ )
            {
                // Trouve pour la piece l'information sur les feeders
                if ( !(strcmp(liste[i].type_feeder, feeder_mv2[k].nom)) )
                {
                    qty_mv2 = feeder_mv2[k].qty;
                    qty_mpa = feeder_mpa[k].qty;
                    espace = feeder_mpa[k].espace;
                    libre_mv2 = ESPACE_MV2 - espace_mv2;
                    libre_mpa = ESPACE_MPA - espace_mpa;
                    if ((espace > libre_mv2)&&(espace > libre_mpa))
                    {
                        break;
                    }
                    else
                    {
                        // Choisi sur quelle machine la piece est posee
                        if ( (qty_mv2 > qty_mpa) && (espace_mv2 <
                            ESPACE_MV2 )&&(libre_mv2 >= espace))
                        {
                            feeder_mv2[k].qty--;
                            espace_mv2 = espace_mv2 + liste[i].espace;
                            strcpy(liste[i].pose_machine,"MV2");
                            liste[i].flag = 1;
                            strcpy(piece_temp, liste[i].nom);
                            mise_a_jour();
                        }
                        else if ( ((qty_mpa) && (espace_mpa < ESPACE_MPA))
                            || !(strcmp(liste[i].type_feeder, "TRAY")) )
                        {
                            feeder_mpa[k].qty--;
                        }
                    }
                }
            }
        }
    }
}

```

```

        espace_mpa = espace_mpa + liste[i].espace;
        strcpy(liste[i].pose_machine, "MPA");
        liste[i].flag = 1;
        strcpy(piece_temp, liste[i].nom);
        mise_a_jour();
    }
    else if ( (qty_mv2) && (espace_mv2 < ESPACE_MV2)
              && (libre_mv2 >= espace) )
    {
        feeder_mv2[k].qty--;
        espace_mv2 = espace_mv2 + liste[i].espace;
        strcpy(liste[i].pose_machine, "MV2");
        liste[i].flag = 1;
        strcpy(piece_temp, liste[i].nom);
        mise_a_jour();
    }
    else if ( (!strncmp(liste[i].type_feeder, "E2412",
                        5)) && (espace_mv2 < ESPACE_MV2) )
    {
        if( feeder_mv2[k+1].qty )
        {
            feeder_mv2[k+1].qty--;
            espace_mv2 = espace_mv2 +
                         liste[i].espace;
            strcpy(liste[i].pose_machine, "MV2");
            liste[i].flag = 1;
            strcpy(piece_temp, liste[i].nom);
            mise_a_jour();
        }
    }
    else if ( (!strncmp(liste[i].type_feeder, "E2416",
                        5)) && (espace_mv2 < ESPACE_MV2) )
    {
        if( feeder_mv2[k-1].qty )
        {
            feeder_mv2[k-1].qty--;
            espace_mv2 = espace_mv2 +
                         liste[i].espace;
            strcpy(liste[i].pose_machine, "MV2");
            liste[i].flag = 1;
            strcpy(piece_temp, liste[i].nom);
            mise_a_jour();
        }
    }
    else
    {
        if ((espace_mv2 == ESPACE_MV2) &&
            (espace_mpa == ESPACE_MPA))
        {
            break;
        }
        else
        {
            printf("\n*** Erreur: Manque d'espace
ou de ressources !!!! ***\n");
            printf(" Pour la piece ->
%s , Ressource -> %s\n", liste[i].nom, liste[i].type_feeder);
            printf(" Ressources MV2 ->
%d , MPA -> %d\n", qty_mv2, qty_mpa);
            printf(" Espaces MV2 ->
%d , MPA -> %d\n", espace_mv2, espace_mpa);
            fprintf(erreur, "%s", "\n***\n");
            Erreur: Manque d'espace ou de ressources !!!! ***\n");
        }
    }
}

```

```

                fprintf(erreur, "                                Pour la
piece  -> %s , Ressource -> %s\n", liste[i].nom, liste[i].type_feeder);      Ressources
                fprintf(erreur, "                                Ressources
MV2 -> %d , MPA -> %d\n", qty_mv2, qty_mpa);                                Espaces
                fprintf(erreur, "                                Espaces
MV2 -> %d , MPA -> %d\n", espace_mv2, espace_mpa);
                }
            }
        }
    }
    global_total += total;
}

//-----
//                                Ecrire_dum ()
//-----
void ecrire_dumgsu()
{
    FILE *fd;
    int z, k, nosub;
    char ptemp[16];

    if ((fd = fopen("dum.txt", "r")) == NULL)
    {
        fprintf(stderr, "Je ne trouve pas le fichier dum.txt\n");
        fprintf(erreur, "%s", "Je ne trouve pas le fichier dum.txt\n");
        exit(-1);
    }

    if ((dum = fopen("dumgsu.txt", "w")) == NULL)
    {
        fprintf(erreur, "%s", "Incapable d' ouvrir le fichier dumgsu.txt\n");
        exit(-1);
    }

    while(fscanf(fd, "%d %s %d", &nosub, ptemp, &k) == 3)
    {
        //fprintf(dum, "Lecture: %d %-16s %d\n", nosub, ptemp, k);
        for (z=0; z<nombre_piece; z++)
        {
            if ((strcmp(liste[z].nom, ptemp)==0)&&(liste[z].flag == 0))
            {
                fprintf(dum, "%d %-16s %d\n", nosub, ptemp, nosub);
            }
        }
        strcpy(ptemp, "");
    }
    fclose(fd);
}

//-----
//                                Main ()
//-----
main()
{
    int i;
}

```

```

if ((erreur = fopen("setup.err", "w+")) == NULL)
{
    fprintf(erreur, "%s", "Incapable d' ouvrir le fichier setup.err\n");
    exit(-1);
}

if ((gsu = fopen("setupgsu.txt", "w")) == NULL)
{
    fprintf(erreur, "%s", "Incapable d' ouvrir le fichier setupgsu.txt\n");
    exit(-1);
}

if ((feed = fopen("feedergs.txt", "w")) == NULL)
{
    fprintf(erreur, "%s", "Incapable d' ouvrir le fichier feedergs.txt\n");
    exit(-1);
}

if ((lis = fopen("listegsu.txt", "w")) == NULL)
{
    fprintf(erreur, "%s", "Incapable d' ouvrir le fichier setupgsu.txt\n");
    exit(-1);
}

// Lit la filiere feeder.txt
lire_feeder();

// Lit la filiere util.txt
lire_utiliser();

// Lit la filiere piece.txt
lire_piece();

// Lit la filiere liste.txt
lire_liste();

global_var = 0;
global_total = 0;

fprintf(gsu, "\nI - SET_UP FIXE MVII=150 espaces MPA=60 espaces ***\n");
fprintf(gsu, "\nParametres - espace declaree MV2 = %d", ESPACE_MV2);
fprintf(gsu, "\n                  - espace declaree MPA = %d", ESPACE_MPA);
fprintf(gsu, "\n                  - espace total      = %d", ESPACE_TOTAL);
fprintf(gsu, "\n                  - nbr_piece_total = %d", nombre_piece);
fprintf(gsu, "\n\nII - LISTE DU SET_UP PERMANENT ***\n");

choisi_piece();

fprintf(gsu, "\n-----PIECE-----MACHINE----FEEDER----ESPACE---\n\n");

for(i = nombre_piece; i > 0; i--)
{
    if(liste[i].flag == 1)
    {
        fprintf(gsu, "%-17s  %-5s  %-6s  %d\n", liste[i].nom,
        liste[i].pose_machine, liste[i].type_feeder, liste[i].espace);
    }
}
fprintf(gsu, "\nEspace MV2 utilise --> %d", espace_mv2);
fprintf(gsu, "\nEspace MPA utilise --> %d", espace_mpa);

fprintf(gsu, "\n\n");

```

```
for (i=0; i<TYPE_FEEDER; i++)
{
    fprintf(feed,"%-7s %-7d %-7d %d\n",feeder_mv2[i].nom,  feeder_mpa[i].qty,
feeder_mv2[i].qty,  feeder_mpa[i].espace);
}
for (i=0; i<nombre_sub; i++)
{
    fprintf(lis,"0 %-3d %-3d 0 %-13s MES      %-16s\n",produit[i].no_sub,
produit[i].qte, produit[i].smt, produit[i].nom);
}
ecrire_dumgsu();
fclose(gsu);
fclose(erreur);
fcloseall();
return 0;
}
```

ANNEXE B

DESCRIPTION DE LA SIMULATION

Voici le rapport echo de Awesim pour la simulation.

AweSim! Input Translator, version 1.4
Copyright (C) 1996 Pritsker Corporation

Reading control NEW ...

```
1 GEN,"Serge Lambert","Regle d'ordonnancement",20 Mars 97,5,YES,YES;
2 INITIALIZE,0,500000,YES;
3 LIMITS,,,15,,,300;
4 NETWORK,READ;
5 FIN;
```

NEW.CON successfully read

Translated file BASECASE.TRA successfully written
Reading network file BASECASE.MNT ...

```
1 #line 0 SMT.NET
1 #line 0 SMT.NET
1      CREATE,INF,0.0, ,1,1;
2      ACTIVITY;
3 SMT_ASSIGN_1: ASSIGN,{{ATRIB[11],2},{ATRIB[10],4},{ATRIB[9],2},
  {ATRIB[8],1},{ATRIB[7],1},{ATRIB[6],1},{ATRIB[5],1}},1;
4      ACTIVITY;
5 TEST1_ASSIGN_1: ASSIGN,{{ATRIB[1],USERF(1)},{II,0}},1;
6      ACTIVITY;
7      ASSIGN,{{ATRIB[14],TNOW}},1;
8      ACTIVITY;
9 PRODUIT: GOON,1;
10     ACTIVITY;
11     ASSIGN,{{II,II+1},{ATRIB[3],USERF(2)}},1;
12     ACTIVITY,1,ATRIB[3]+ATRIB[4];
13 SUIVANT: GOON,2;
14     ACTIVITY, , ,II < ATRIB[1];
15     ACTIVITY, , , "SMT_COLCT_2";
16     ACTIVITY, , , "SMT_GOON_1";
17     ASSIGN,{{atrib[12],atrib[12] - atrib[2]}},1;
18     ACTIVITY, , , "PRODUIT";
19 SMT_COLCT_2: COLCT, ,ATRIB[1],"Nombre de commande", , , ,1;
20     ACTIVITY;
21     COLCT, ,ATRIB[2],"Quantite produite", , , ,1;
22     ACTIVITY;
23     COLCT, ,ATRIB[3],"Temps de setup", , , ,1;
24     ACTIVITY;
```

```

25      COLCT, , ATRIB[4], "Temps de production", , , ,1;
26      ACTIVITY;
27      COLCT, , TNOW - ATRIB[14], "Temps de passage", , , ,1;
28      ACTIVITY;
29      COLCT, , ATRIB[12], "WIP", , , ,1;
30      ACTIVITY, , , , "SMT_TERMINATE_1";
31 SMT_TERMINATE_1: TERMINATE, INF;
32 SMT_GOON_1: GOON, 2;
33      ACTIVITY, , , , "TEST1_ASSIGN_1";
34      ACTIVITY, , , , "Setup";
35 Setup: COLCT, , ATRIB[13], "Setup de la semaine", , , ,1;
36      ACTIVITY;
37 Ratio: COLCT, , ATRIB[15], "Ratio Setup/Setup+Production", , , ,1;
38      ACTIVITY, , , , "SMT_TERMINATE_1";
39      END;

```

BASECASE.MNT successfully read

Translated network file BASECASE.TRN successfully written

Le fichier intermédiaire de la simulation:

AweSim! Simulator, version 1.4
Copyright (C) 1996 Pritsker Corporation

Translated file BASECASE.TRA successfully read

```

** AweSim! Version 1.4 ECHO REPORT **
Tue Mar 25 18:02:18 1997

```

Simulation Project : Regle d'ordonnancement
Modeler : Serge Lambert
Date : 20 MARS 97

Run options

```

Run number 1 of 5
Beginning time of run : 0.000000
Ending time of run    : 500000.000000
Maximum errors during run : 1
Maximum entities in system : 300
Clear statistics between runs : YES
Execute simulation after input: YES
Warn of destroyed entities   : YES
Generate summary report : EVERY(1)

```

Variables

```

Number of LL variables : 0
Number of XX variables : 0
Number of SZ variables : 0
Number of entity ATRIBs: 16
Number of entity LTRIBs: 0
Number of entity STRIBs: 0

```

Collect Information

COLCT NUMBER	IDENTIFIER	HISTOGRAM SPECIFICATIONS		
		NCEL	HLOW	HWID
1	Nombre de commandes	0	0.0000	0.0000
2	Quantité produit	0	0.0000	0.0000
3	Temps de setup	0	0.0000	0.0000
4	Temps de product	0	0.0000	0.0000
5	Temps de passage	0	0.0000	0.0000
6	WIP	0	0.0000	0.0000
7	Setup de la sema	0	0.0000	0.0000
8	Ratio Setup/Set	0	0.0000	0.0000

Random Number Streams Information

STREAM NUMBER	SEED VALUE	REINITIALIZATION OF STREAM
1	428956419	NO
2	1954324947	NO
3	1145661099	NO
4	1835732737	NO
5	794161987	NO
6	1329531353	NO
7	200496737	NO
8	633816299	NO
9	1410143363	NO
10	1282538739	NO
11	794026294	NO
12	977821281	NO
13	699858332	NO
14	1683733431	NO
15	96358723	NO
16	602885281	NO
17	871633513	NO
18	1984612552	NO
19	232391877	NO
20	135618479	NO
21	1814383333	NO
22	324184021	NO
23	1667630903	NO
24	1082038467	NO
25	1885999711	NO
26	111454913	NO
27	1262746665	NO
28	725514806	NO
29	2087308334	NO
30	1027457115	NO
31	1716773784	NO
32	1417840845	NO
33	1401170757	NO
34	505042365	NO
35	1554339362	NO
36	1747494519	NO
37	93505551	NO
38	194910479	NO
39	1297383121	NO
40	806401626	NO

41	1242558033	NO
42	1722906649	NO
43	1195299681	NO
44	57081438	NO
45	817128895	NO
46	1919094954	NO
47	1435250780	NO
48	1278785392	NO
49	1309679730	NO
50	1281484595	NO
51	394601628	NO
52	85886326	NO
53	1571729619	NO
54	918904909	NO
55	1864761823	NO
56	1268956513	NO
57	1034484093	NO
58	1763605842	NO
59	1039242913	NO
60	1322164949	NO
61	1783293536	NO
62	70884048	NO
63	1420035359	NO
64	994868838	NO
65	1463578868	NO
66	790460117	NO
67	1431310689	NO
68	268966977	NO
69	1446647089	NO
70	1990098112	NO
71	893770676	NO
72	756136149	NO
73	1758938931	NO
74	1098677441	NO
75	1178898083	NO
76	271054147	NO
77	1009598300	NO
78	2114861268	NO
79	1617952069	NO
80	1131840250	NO
81	2094070757	NO
82	195319339	NO
83	1659214966	NO
84	1440396285	NO
85	563832118	NO
86	1702510512	NO
87	1721643437	NO
88	2054527950	NO
89	231008758	NO
90	811760922	NO
91	931421337	NO
92	2056682338	NO
93	1224747865	NO
94	2087343551	NO
95	434301072	NO
96	1002794063	NO
97	1132277789	NO
98	1505059305	NO

99	208486230	NO
100	1769772683	NO

Intermediate results for run 1

Maximum number of entities concurrently in system is 3
Intermediate results for run 2

Maximum number of entities concurrently in system is 3
Intermediate results for run 3

Maximum number of entities concurrently in system is 3
Intermediate results for run 4

Maximum number of entities concurrently in system is 3
Intermediate results for run 5

Maximum number of entities concurrently in system is 3

0 total errors during execution

La fonction Userf sert à générer les commandes, aller lire les bases de données pour les produits, les ordonner et simuler. La figure 4 montre les interrelations entre la simulation et les programmes décrits à l'annexe 1. La fonction Userf(1) fait appelle à un exécutable DOS (batchfile) pour exécuter les différents programmes nécessaires à la simulation. La fonction Userf(2) est utilisée durant la simulation trouver quelle commande doit être assemblée. Le code est présenté dans les prochaines pages.

Voici le fichier gsur2x30.bat. Il sert pour les simulations de la politique GSU, la réduction de pièces et les cassettes doubles lorsque rho est à 30%. La première partie est consacrée à effacer les fichiers de la précédente génération de familles. De plus, il remplace les fichiers pour les cassettes avec ceux contenant l'information pour les cassettes doubles. Il sert à copier, effacer et appeler des programmes nécessaires pour obtenir le fichier contenant les familles de produits.

```
del setup.txt
del statut.err
del dum.txt
del liste.txt
del feeder.txt
del util.txt
copy feeder2x.bak feeder.txt
gentxt3 %1 > statut.err
pieces
del dum.bak
del liste.bak
ren dum.txt dum.bak
ren liste.txt liste.bak
ren dumred.txt dum.txt
ren listedred.txt liste.txt
util >> statut.err
gsu2 >> statut.err
del feeder.txt
ren feedergs.txt feeder.txt
del dum.txt
ren dumgsu.txt dum.txt
del liste.txt
ren listegsu.txt liste.txt
sds1x30 >> statut.err
```

Code de la fonction Userf dans le réseau SLAMII de la simulation.

```
*****
Simulation du systeme de Production

Programme: userf.c
Fonction: Interface C pour SLAM
Auteur: Serge Lambert
Date: 19 octobre 1996
*****
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <malloc.h>
#include <math.h>
#include <windows.h>
#include <time.h>
#include "vslam.h"
#include "maitrise.h"

int Var5, Var6, Var7, Var8, Var9, Var10, Var11;
int MAXPROD;
int DiffProd;
double SetupSemaine;
double TempsProd;
double WIP;
GLOBAL hprod, hdon, hsum, hsup;

*****
Fonction: USERF()

*****
double SWFUNC USERF (int iCode, ENTITY *peUser)
{
    double donnee;
    double commande;
    int n, i, j, k;
    int ligne;
    char ltemp[2];
    char string[128];
    char buffer[16];
    char scenario[18];
    char *token;
    char espace[] = ",";
    double no_al, temp ;
    PRODUIT *prod, *ptrprod;
    FILE *fin1_ptr, *fin2_ptr, *fout1_ptr, *fout2_ptr, *fout4_ptr;
    char *argv1 = "produit.txt";
    char *argv2 = "setup.txt";
    char *argv3 = "test.res";
    char *argv4 = "simulate.txt";
    char *argv6 = "ligne.txt";
    double Round();
    double Sup();
    void ylire(FILE *, FILE *);
    void OrdreFam(FILE *, FILE *);

    switch (iCode)
```

```

{
case 1:
    Var5 = (int) peUser->ATTRIB[5];
    Var6 = (int) peUser->ATTRIB[6];
    Var7 = (int) peUser->ATTRIB[7];
    Var8 = (int) peUser->ATTRIB[8];
    Var9 = (int) peUser->ATTRIB[9];
    Var10 = (int) peUser->ATTRIB[10];
    Var11 = (int) peUser->ATTRIB[11];

    /* ouverture et ecriture du fichier indicant la ligne */
    (fout4_ptr = fopen (argv6, "w"));
    fprintf(fout4_ptr,"1 \n");
    fclose(fout4_ptr);

    commande = Round(RNORM(33, 9, 8));

    /* Troncage si commande inferieur a 0 */
    if (commande<0)
    {
        commande = 8;
        Beep(500,500);
    }

    donnee = commande;
    MAXPROD = (int) commande ;

    /* Allocation memoire pour les nombres aleatoires */
    hprod = GlobalAlloc(GHND, sizeof(PRODUIT)*MAXPROD);
    prod = (PRODUIT *) GlobalLock(hprod);

    /* Generation des nombres aleatoires */
    j = 0;
    for (n=0; n<MAXPROD; n++)
    {
        no_al = UNFRM(0., 2827.,9);

        /* Ecriture et trie en bulle des nombres */
        if (j==0)
        {
            prod->chiffre = no_al;
        }
        else if (j>0)
        {
            k = j;
            ptrprod = prod;
            for (i = 0; i < j; i++)
            {
                prod--;
                k--;
                if (no_al < prod->chiffre)
                {
                    temp = prod->chiffre;
                    prod++;
                    k++;
                    prod->chiffre = temp;
                    prod--;
                    k--;
                    if (k==0) prod->chiffre = no_al;
                }
            }
            else
            {
                prod++;
                k++;
            }
        }
    }
}

```

```

        prod->chiffre = no_al;
    }
}
prod = ptrprod;
}//fin du else if
j++;
prod++;
} //fin du for n

/* Liberation de la memoire */
GlobalUnlock(hprod);

/* ouverture du fichier d'entree */
(finl_ptr = fopen (argv1, "r"));

/* ouverture des fichiers de sortie */
(foutl_ptr = fopen (argv3, "w"));

/* Lecture du fichier produit et ecriture du fichier pour famille */
y1ire (finl_ptr, foutl_ptr);

/* fermeture du fichier d'entree et de sortie */
fclose (finl_ptr);
fclose (foutl_ptr);

if (Var5 > 0)
{
    /* Generation des familles a partir de test.res */
    /* Choix formation des familles */
    if (Var5 == 1) strcpy(scenario, "sds");
    else strcpy(scenario, "gsu");
    /* Choix reduction de pieces */
    if (Var8 == 1) strcat(scenario, "0");
    else strcat(scenario, "r");
    /* Choix des feeders */
    if (Var6 == 1) strcat(scenario, "1x");
    else strcat(scenario, "2x");
    /* Choix du niveau de similarite */
    if (Var7 == 1) strcat(scenario, "10");
    else strcat(scenario, "30");
    strcat(scenario, " test.res");
    system(scenario);
}

/* ouverture des fichiers d'entree pour les familles*/
(fin2_ptr = fopen (argv2, "r"));

/* ouverture du fichier de sortie pour les familles*/
(fout2_ptr = fopen (argv4, "w"));

/* Appel fonction pour ordonner les familles */
OrdreFam(fout2_ptr, fin2_ptr);

/* fermeture des fichiers d'entree et de sortie */
fclose (fin2_ptr);
fclose (fout2_ptr);
peUser->ATRIB[12] = WIP;
peUser->ATRIB[13] = SetupSemaine;
peUser->ATRIB[15] = (SetupSemaine / (SetupSemaine + TempsProd));

break;

case 2:

```

```

/* lire la ligne rendu */
(fout4_ptr = fopen(argv6, "r"));
while (!feof (fout4_ptr))
{
    fscanf(fout4_ptr, "%s", ltemp);
    ligne = atoi(ltemp);
}
fclose (fout4_ptr);

/* lecture du temps de setup du produit a faire */
(fout2_ptr = fopen (argv4, "r"));
i = 1;
while (!feof (fout2_ptr))
{
    fgets(string, 128, fout2_ptr);
    token = strtok (string, espace);
    n = 0;
    while (token != NULL)
    {
        strcpy(buffer, token);
        if (i == ligne)
        {
            switch (n)
            {
                case 1:
                    peUser->ATTRIB[2] = atof(buffer);
                    break;
                case 2:
                    donnee = atof(buffer);
                    break;
                case 3:
                    peUser->ATTRIB[4] = atof(buffer);
                    break;
            }
        }
        n++;
        token = strtok(NULL, espace);
    }
    i++;
}
fclose(fout2_ptr);

/* ecriture de la ligne suivante */
ligne++;
if((fout4_ptr = fopen(argv6, "w"))==NULL)
{
    Beep(500,500);
}
fprintf(fout4_ptr, "%d \n", ligne);
fclose (fout4_ptr);
break;

} //fin du case
return (donnee);
}
/* fin de SWUSERF */

*****Fonction: Lecture du fichier d'entree*****void ylire (fin1_ptr, fout1_ptr)

```

```

FILE *fin1_ptr, *fout1_ptr;
{
    int i, j;
    int n, m, p, t;
    char code_prod[16] = "";
    char fichierSMT[13] = "";
    char compagnie[4] = "";
    double lot_cum = 0;
    double cumul;
    double moy;
    double ecart;
    double ttop1;
    double ttop2;
    double tbot;
    double quantite;
    char string[128];
    char virgule[] = ", ";
    char *token;
    char buffer[16];
    PRODUIT *prod;
    DONNEE *don;
    FILE *PasFam;

    /* parcourir le fichier d'entree */
    fscanf (fin1_ptr, "%[^\\n]", string);
    i = 0;
    n = 1; p = 1;
    m = 0;
    DiffProd = 0;
    prod = (PRODUIT *)GlobalLock(hprod);
    hdon = GlobalAlloc(GHND, sizeof(DONNEE)*MAXPROD);
    don = (DONNEE *)GlobalLock(hdon);
    cumul = prod->chiffre;
    if (Var5 == 0) PasFam = fopen("setup.txt", "w");
    while (!feof (fin1_ptr))
    {
        fgets(string, 128, fin1_ptr);
        token = strtok(string, virgule);
        j = 0;
        while (token != NULL)
        {
            strcpy(buffer,token);
            if (i>0)
            {
                switch (j)
                {
                    case 0:
                        strcpy(code_prod,buffer);
                        break;
                    case 1:
                        strcpy(fichierSMT,buffer);
                        break;
                    case 2:
                        strcpy(compagnie,buffer);
                        break;
                    case 3:
                        lot_cum = atoi(buffer);
                        while ((lot_cum>cumul)&&(n<MAXPROD+1))
                        {
                            fprintf (fout1_ptr, "%-16s", code_prod);
                            fprintf (fout1_ptr, "%-14s", fichierSMT);
                            fprintf (fout1_ptr, "%3s\\n", compagnie);
                            if (Var5 == 0) fprintf (PasFam, "1      %-14s\\n", fichierSMT);
                            prod++;
                        }
                }
            }
        }
    }
}

```

```

        cumul = prod->chiffre;
        m++;
        n++;
    }
    break;
case 4:
    moy = atof(buffer);
    break;
case 5:
    ecart = atof(buffer);
    break;
case 6:
    ttop1 = atof(buffer);
    break;
case 7:
    ttop2 = atof(buffer);
    break;
case 8:
    tbot = atof(buffer);
    if (m>0)
    {
        DiffProd++;
        for (t = 0; t < m; t++)
        {
            if (ecart !=0)
            {
                quantite = Round(RNORM(moy,ecart,3));
                if (quantite<1) quantite = 5;
            }
            else quantite = moy;

            strcpy(don->nom, code_prod);
            strcpy(don->fichier, fichiersSMT);
            don->quant = quantite;
            don->top1 = Sup(quantite*ttop1/42);
            don->top2 = Sup(quantite*ttop2/42);
            don->bottom = Sup(quantite*tbot/42);
            don++;
        }
    }
    m = 0;
    break;
} //fin du switch
} //fin du if

j++;
token = strtok(NULL, virgule);
} //fin du second while
i++;
} //fin du premier while
if (Var5 == 0) fclose(PasFam);
GlobalUnlock(hprod);
GlobalFree(hprod);
GlobalUnlock(hdon);
}
/* fin de ylire */

*****
Fonction: Arrondir un nombre
*****
double Round (double number)

```

```

{
    double chiffre;
    double entier;

    if (modf(number, &entier)>.5) chiffre = ceil(number);
    else chiffre=entier;

    return chiffre;
}
/* fin de Round */

*****Fonction: Arrondir un nombre a l'entier superieur*****Fonction: Arrondir un nombre a l'entier superieur

double Sup (double number)
{
    double chiffre;
    double entier;

    if (modf(number, &entier)>.2) chiffre = ceil(number);
    else chiffre=entier;

    return chiffre;
}
/* fin de Sup */

*****Fonction: Ordonnancement des familles*****Fonction: Ordonnancement des familles

void OrdreFam(fout2_ptr, fin2_ptr)
FILE  *fout2_ptr, *fin2_ptr;
{
    int i, j, k, l, m;
    char  nofam[1];
    char  fichierSMT[13];
    char  *pro, *tprod;
    int   total_famille;
    int   PasChoisi;
    double tempol, tempo2, tempo3;
    DONNEE  *don, *ptrtemp1;
    SOMME   *sum, *ptrtemp2;
    SUPER   *sup, *ptrtemp4;
    char   tempfich[13];
    double tqte, ttop1, ttop2, tbot;
    double temptop[100], tempbot[100];
    int    batch[100], ordre[100];
    int    tfam, tord, bidon, QuantiteSetup;

    hsum = GlobalAlloc(GHND, sizeof(SOMME)*MAXPROD);
    sum = (SOMME *)GlobalLock(hsum);
    hsup = GlobalAlloc(GHND, sizeof(SUPER)*MAXPROD);
    sup = (SUPER *)GlobalLock(hsup);
    don = (DONNEE *)GlobalLock(hdon);
    ptrtemp1 = don;
    i = 0;

    /* Incrire famille dans tableau */
    while (!feof (fin2_ptr))
    {
        fscanf(fin2_ptr, "%s %s", nofam, fichierSMT);

```

```

if (i<MAXPROD)
{
    don = ptrtemp1;
    k = 0;
    while (k < MAXPROD)
    {
        if ((i==0)&&(strcmp(don->fichier,fichierSMT)==0))
        {
            strcpy (sup->produit, don->fichier);
            sup->qte = don->quant;
            sup->ttl = don->top1;
            sup->tt2 = don->top2;
            sup->tb = don->bottom;
            sup->groupe = atoi(nofam);
            don->famille = atoi(nofam);
            sup->arrive = i+1;
            sup->seqprod = 0;
            i++;
            sup++;
            break;
        }
        else if ((i>0)&&(strcmp(don->fichier,fichierSMT)==0)&&(don->famille==0))
        {
            ptrtemp4 = sup;

            for (j = 0; j < i+1 ; j++)
            {
                sup--;
                if (strcmp(sup->produit,don->fichier) == 0)
                {
                    if (sup->ttl > don->top1)
                    {
                        tqte = sup->qte;
                        ttop1 = sup->ttl;
                        ttop2 = sup->tt2;
                        tbot = sup->tb;
                        tprod = sup->produit;
                        tfam = sup->groupe;
                        tord = sup->arrive;
                        sup++;
                        sup->qte = tqte;
                        sup->ttl = ttop1;
                        sup->tt2 = ttop2;
                        sup->tb = tbot;
                        strcpy(sup->produit, tprod);
                        sup->groupe = tfam;
                        sup->arrive = tord;
                        sup->seqprod = 0;
                        sup--;
                    }
                }
                else
                {
                    sup++;
                    strcpy (sup->produit, don->fichier);
                    sup->qte = don->quant;
                    sup->ttl = don->top1;
                    sup->tt2 = don->top2;
                    sup->tb = don->bottom;
                    sup->groupe = atoi(nofam);
                    don->famille = atoi(nofam);
                    sup->arrive = i+1;
                    sup->seqprod = 0;
                }
            }
        }
    }
}

```

```

if (strcmp(sup->produit,don->fichier) != 0)
{
    sup++;
    strcpy (sup->produit, don->fichier);
    sup->qte = don->quant;
    sup->ttl = don->top1;
    sup->tt2 = don->top2;
    sup->tb = don->bottom;
    sup->choisi = 0;
    sup->groupe = atoi(nofam);
    don->famille = atoi(nofam);
    sup->arrive = i+1;
    sup->seqprod = 0;
}
} // fin du for
sup = ptrtemp4;
i++;
sup++;
break;
} //fin du elseif
k++;
don++;
}//fin du while k
} //fin du if i
total_famille = atoi(nofam);
QuantiteSetup = i;
} //fin du while

/* Si produit avec seulement bottom ou erreur dans programme famille */
if (MAXPROD != QuantiteSetup)
{
    don = ptrtemp1;
    for ( i = 0; i < MAXPROD; i++, don++)
    {
        if ((don->famille < 1) || (don->famille > total_famille))
        {
            don->famille = 0;
            strcpy (sup->produit, don->fichier);
            sup->ttl = don->top1;
            sup->tt2 = don->top2;
            sup->tb = don->bottom;
            sup->groupe = 0;
            sup++;
        }
    }
}
GlobalUnlock(hsup);

/* Regroupement par produit et ordonner selon SPT*/
if (Var9 == 1)
{
    sup = (SUPER *)GlobalLock(hsup);
    i=0;
    j=1;
    for (i=0; i<MAXPROD; i++)
    {
        tempol = sup->ttl;
        tempo2 = sup->tt2;
        tempo3 = sup->tb;
        bidon = sup->groupe;
        pro = sup->produit;
        /* Trie en bulle des produits */
        if (j==1)
        {

```

```

sum->stop1 = tempol;
sum->stop2 = tempo2;
sum->sbot = tempo3;
sum->family = bidon;
strcpy(sum->fich, pro);
}
else if (j>1)
{
ptrtemp2 = sum;
for (k = 1; k < j+1; k++)
{
sum--;
if (sum->stop1 > tempol)
{
ttop1 = sum->stop1;
ttop2 = sum->stop2;
tbot = sum->sbot;
tprod = sum->fich;
tfam = sum->family;
sum++;
sum->stop1 = ttop1;
sum->stop2 = ttop2;
sum->sbot = tbot;
strcpy(sum->fich, tprod);
sum->family = tfam;
sum--;
}
else
{
sum++;
sum->stop1 = tempol;
sum->stop2 = tempo2;
sum->sbot = tempo3;
sum->family = bidon;
strcpy(sum->fich, pro);
}
}
// fin du for
sum = ptrtemp2;
} //fin du elseif
sum++;
j++;
sup++;
}
}
else if (Var9 == 2)
{
sup = (SUPER *)GlobalLock(hsup);
i=0;
j=1;
for (i=0; i<MAXPROD; i++)
{
PasChoisi = 1;
if(sup->choisi == 0)
{
PasChoisi = 0;
pro = sup->produit;
bidon = sup->groupe;
tempol = sup->tt1;
tempo2 = sup->tt2;
tempo3 = sup->tb;
ptrtemp4 = sup;
sup++;
for(l=1; l<MAXPROD-i; sup++, l++)
{

```

```

        if(strcmp(sup->produit, pro) == 0)
        {
            tempol += sup->ttl;
            tempo2 += sup->tt2;
            tempo3 += sup->tb;
            sup->choisi = 1;
        }
    }
    sup = ptrtemp4;
}
/* Trie en bulle des produits */
if ((j==1)&&(PasChoisi==0))
{
    sum->stop1 = tempol;
    sum->stop2 = tempo2;
    sum->sbot = tempo3;
    sum->family = bidon;
    strcpy(sum->fich, pro);
}
else if ((j>1)&&(PasChoisi==0))
{
    ptrtemp2 = sum;
    for (k = 1; k < j+1; k++)
    {
        sum--;
        if (sum->stop1 > tempol)
        {
            ttop1 = sum->stop1;
            ttop2 = sum->stop2;
            tbot = sum->sbot;
            tprod = sum->fich;
            tfam = sum->family;
            sum++;
            sum->stop1 = ttop1;
            sum->stop2 = ttop2;
            sum->sbot = tbot;
            strcpy(sum->fich, tprod);
            sum->family = tfam;
            sum--;
        }
        else
        {
            sum++;
            sum->stop1 = tempol;
            sum->stop2 = tempo2;
            sum->sbot = tempo3;
            sum->family = bidon;
            strcpy(sum->fich, pro);
        }
    } // fin du for
    sum = ptrtemp2;
} //fin du elseif
if(PasChoisi == 0)
{
    sum++;
    j++;
}
sup++;
} //fin du for
}
GlobalUnlock(hsum);
GlobalUnlock(hsup);
DiffProd = j-1;

```

```

/* Ordre des produits */
if (Var11 == 1) /* Ordre aleatoires */
{
    sum = (SOMME *)GlobalLock(hsum);
    sup = (SUPER *)GlobalLock(hsup);
    ptrtemp2 = sum;
    ptrtemp4 = sup;
    tfam = 1;
    bidon = 0;
    i = 0;
    while (i<MAXPROD)
    {
        k = 0;
        bidon = (int) UNFRM(0., MAXPROD, 2);
        sum = ptrtemp2;
        sup = ptrtemp4;
        for (j=0; j<bidon; j++, sum++);
        for (l=0; l<MAXPROD; l++, sup++)
        {
            if ((sum->family == sup->groupe)&&(sup->seqprod > 0))
            {
                if (sup->seqprod > k) k = sup->seqprod;
            }
            sup = ptrtemp4;
            for (l=0; l<MAXPROD; l++, sup++)
            {
                if (Var9 == 1)
                {
                    if((strcmp(sum->fich, sup->produit) == 0)&&(sum->stop1 == sup->ttl))
                    {
                        if (sup->seqprod == 0)
                        {
                            sup->seqprod = k + 1;
                            i++;
                            break;
                        }
                    }
                }
                else
                {
                    if(strcmp(sum->fich, sup->produit) == 0)
                    {
                        if (sup->seqprod == 0)
                        {
                            sup->seqprod = k + 1;
                            k++;
                            i++;
                        }
                    }
                }
            } //fin du if meme produit
        } //fin du for j
    }
}
if (Var11 == 2) /* Ordre SPT */
{
    sum = (SOMME *)GlobalLock(hsum);
    sup = (SUPER *)GlobalLock(hsup);
    ptrtemp2 = sum;
    ptrtemp4 = sup;
    tfam = 1;
    for (i=0; i<total_famille; i++)
    {
        k = 1;

```

```

sum = ptrtemp2;
for (j=0; j<DiffProd; j++, sum++)
{
    if (sum->family == tfam)
    {
        sup = ptrtemp4;
        for (l=0; l<MAXPROD; l++, sup++)
        {
            if (Var9 == 1)
            {
                if((strcmp(sum->fich, sup->produit) == 0)&&(sum->stop1 == sup->tt1))
                {
                    if (sup->seqprod == 0)
                    {
                        sup->seqprod = k;
                        k++;
                    }
                }
            }
            else
            {
                if(strcmp(sum->fich, sup->produit) == 0)
                {
                    sup->seqprod = k;
                    k++;
                }
            }
        } //fin du if meme produit
    } // fin du for l
} // fin du if meme famille
} //fin du for j
tfam++;
}
}
GlobalFree(hsum);

/* temps de setup */
tfam = 1;
m = 0;
{
    for (i=0; i<total_famille; i++)
    {
        k = 1;
        for (j=0; j<MAXPROD; j++)
        {
            sup = ptrtemp4;
            for (l=0; l<MAXPROD; l++, sup++)
            {
                if ((sup->groupe == tfam)&&(sup->seqprod == k))
                {
                    if (k == 1)
                    {
                        if (Var5==2) sup->tsetup = 35.0;
                        else sup->tsetup = 75.0;
                        strcpy(tempfich, sup->produit);
                        m = 1;
                    }
                    if (k > 1)
                    {
                        if (strncmp(tempfich, sup->produit,4) == 0 )
                        {
                            if (strcmp(tempfich, sup->produit) == 0 ) sup->tsetup = 0.0;
                            else sup->tsetup = 15.0;
                        }
                    }
                }
            }
        }
    }
}

```

```

        {
            if (Var5 == 0) sup->tsetup = 75.0;
            else sup->tsetup = 25.0;
        }
        strcpy(tempfich, sup->produit);
    }
    k++;
}
}
tfam++;
}
}
GlobalUnlock(hsup);

/* Choix de l'ordre des familles selon SPT */
sup = (SUPER *)GlobalLock(hsup);
ptrtemp4 = sup;
for (i = 1; i < total_famille+1; i++)
{
    tempol = 0;
    tempo2 = 0;
    for(j = 0; j < MAXPROD; j++, sup++)
    {
        if (sup->groupe == i)
        {
            if (Var10 == 1) tempol = i;
            if (Var10 == 2) tempol += sup->ttl;
            if (Var10 == 3) tempol += sup->tsetup;
            if (Var10 == 4) tempol += sup->ttl + sup->tsetup;
            tfam = sup->groupe;
        }
    }
}

/* Trie en bulle des familles */
if (i==1)
{
    temptop[i] = tempol;
    tempbot[i] = tempo2;
    batch[i] = tfam;
}
else if (i>1)
{
    k = i;

    for (j = 1; j < k+1; j++)
    {
        i--;
        if (temptop[i] > tempol)
        {
            ttop1 = temptop[i];
            tbot = tempbot[i];
            bidon = batch[i];
            i++;
            temptop[i] = ttop1;
            tempbot[i] = tbot;
            batch[i] = bidon;
            i--;
            if (i == 1)
            {
                temptop[i] = tempol;
                tempbot[i] = tempo2;
                batch[i] = tfam;
            }
        }
    }
}

```

```

    }
    else
    {
        i++;
        temptop[i] = tempol;
        tempbot[i] = tempo2;
        batch[i] = tfam;
    }
} // fin du for
i = k;
} // fin du elseif
ordre[i] = i;
sup = ptrtemp4;
} // fin du for
GlobalUnlock(hsup);

/* Ecrit ordre des familles */
sup = (SUPER *)GlobalLock(hsup);
ptrtemp4 = sup;
for (i=0; i<total_famille; i++)
{
    for (l=1; l<total_famille+1; l++)
    {
        if (ordre[l] == i+1)
        {
            sup = ptrtemp4;
            for (j=0; j<MAXPROD; j++, sup++)
            {
                if (sup->groupe == batch[l]) sup->seqfam = ordre[l];
            }
        }
    }
}

/* Ecriture au fichier simulate.txt */
tfam = 1;
WIP = 0.0;
SetupSemaine = 0.0;
TempsProd = 0.0;
m = 0;
{
    for (i=0; i<total_famille; i++)
    {
        k = 1;
        for (j=0; j<MAXPROD; j++)
        {
            sup = ptrtemp4;
            for (l=0; l<MAXPROD; l++, sup++)
            {
                if ((sup->seqfam == tfam)&&(sup->seqprod == k))
                {
                    if (k == 1) /* Correction temps de setup GSU */
                    {
                        if ((Var5==2)&&(m==0)) sup->tsetup = 75.0;
                        m = 1;
                    }
                    SetupSemaine += sup->tsetup;
                    TempsProd += sup->ttl + sup->tt2;
                    WIP += sup->qte;
                    fprintf(fout2_ptr, "%s,%f,%f,%f,%f,%f,%d,%d\n", sup->produit, sup->qte, sup->tsetup, sup->ttl +
                        sup->tt2, sup->tt2, sup->tb, sup->groupe, sup->seqfam, sup->seqprod);
                    k++;
                }
            }
        }
    }
}

```

```
        }
    }
    tfam++;
}
GlobalUnlock(hsup);
GlobalFree(hsup);
GlobalUnlock(hdon);
GlobalFree(hdon);
*/
/* fin de Ordrefam */
```

ANNEXE C

SAS EXPÉRIENCE 1

Voici le programme SAS pour l'analyse de variance et l'étude de Duncan des variables.

```
data matrox;
  options pagesize=60 linesize=80;
  options nodate;
  title;
  title5 'Simulation Ligne SMT';
input methodes$ feeder rho reduct$ interf$ intraf$ wip tpas setup;
cards;
SDS 150 0.10 AUCUNE RANDOM RANFAM 633.4776423 1265.261726 935.4314721
SDS 150 0.10 AUCUNE RANDOM RANFAM 621.070429 1214.68923 947.3880597
SDS 150 0.10 AUCUNE RANDOM RANFAM 627.947505 1228.602777 938.275
SDS 150 0.10 AUCUNE RANDOM RANFAM 622.4484024 1234.755848 943.7
SDS 150 0.10 AUCUNE RANDOM RANFAM 606.1992205 1205.496477 947.0689655
SDS 150 0.10 AUCUNE RANDOM SPTFAM 662.6128221 1230.812539 948.3163265
SDS 150 0.10 AUCUNE RANDOM SPTFAM 646.8720735 1184.313543 959.425
SDS 150 0.10 AUCUNE RANDOM SPTFAM 651.4150161 1202.302933 954.3467337
SDS 150 0.10 AUCUNE RANDOM SPTFAM 649.43497 1210.006618 958.1909548
SDS 150 0.10 AUCUNE RANDOM SPTFAM 629.8338359 1183.393697 960.0990099
SDS 150 0.10 AUCUNE SPT RANFAM 781.018142 1197.160932 935.4314721
SDS 150 0.10 AUCUNE SPT RANFAM 745.509577 1172.841528 947.3880597
SDS 150 0.10 AUCUNE SPT RANFAM 764.7377574 1182.496415 938.275
SDS 150 0.10 AUCUNE SPT RANFAM 751.6641186 1192.002293 943.7
SDS 150 0.10 AUCUNE SPT RANFAM 719.7159994 1176.489729 947.0689655
SDS 150 0.10 AUCUNE SPT SPTFAM 809.6512724 1162.312284 948.3163265
SDS 150 0.10 AUCUNE SPT SPTFAM 771.1928779 1143.44689 959.425
SDS 150 0.10 AUCUNE SPT SPTFAM 786.6513663 1155.867516 954.3467337
SDS 150 0.10 AUCUNE SPT SPTFAM 778.9813674 1166.079612 958.1909548
SDS 150 0.10 AUCUNE SPT SPTFAM 742.7270946 1153.295961 960.0990099
SDS 150 0.10 RED20% RANDOM RANFAM 629.9177591 1271.606898 929.2893401
SDS 150 0.10 RED20% RANDOM RANFAM 615.4349871 1222.02382 941.3681592
SDS 150 0.10 RED20% RANDOM RANFAM 623.9893325 1238.529869 930.625
SDS 150 0.10 RED20% RANDOM RANFAM 617.5079438 1246.131836 940.35
SDS 150 0.10 RED20% RANDOM RANFAM 600.287358 1216.127765 941.6666667
SDS 150 0.10 RED20% RANDOM SPTFAM 660.0271373 1238.017882 943.3163265
SDS 150 0.10 RED20% RANDOM SPTFAM 642.8687548 1193.04767 954.475
SDS 150 0.10 RED20% RANDOM SPTFAM 652.1431415 1205.440447 946.825
SDS 150 0.10 RED20% RANDOM SPTFAM 644.8989697 1219.395664 955.3768844
SDS 150 0.10 RED20% RANDOM SPTFAM 625.0558651 1191.640416 955.8910891
SDS 150 0.10 RED20% SPT RANFAM 778.4943855 1200.915003 929.2893401
SDS 150 0.10 RED20% SPT RANFAM 742.7263078 1170.97392 941.3681592
SDS 150 0.10 RED20% SPT RANFAM 761.1008531 1189.46557 930.625
SDS 150 0.10 RED20% SPT RANFAM 751.6523002 1194.113862 940.35
SDS 150 0.10 RED20% SPT RANFAM 715.367902 1184.787657 941.6666667
SDS 150 0.10 RED20% SPT SPTFAM 808.1960169 1167.081857 943.3163265
SDS 150 0.10 RED20% SPT SPTFAM 770.2935164 1144.287109 954.475
```

SDS	150	0.10	RED20%	SPT	SPTFAM	788.5140759	1156.559976	946.825
SDS	150	0.10	RED20%	SPT	SPTFAM	779.3380477	1166.491007	955.3768844
SDS	150	0.10	RED20%	SPT	SPTFAM	740.2188441	1158.75587	955.8910891
SDS	150	0.30	AUCUNE	RANDOM	RANFAM	604.5252251	1472.566503	1146.961326
SDS	150	0.30	AUCUNE	RANDOM	RANFAM	585.0024855	1426.456007	1156
SDS	150	0.30	AUCUNE	RANDOM	RANFAM	586.6649527	1444.899154	1146
SDS	150	0.30	AUCUNE	RANDOM	RANFAM	583.6038397	1448.049416	1160.679348
SDS	150	0.30	AUCUNE	RANDOM	RANFAM	558.9965725	1415.44916	1154.126984
SDS	150	0.30	AUCUNE	RANDOM	SPTFAM	632.3938981	1444.99233	1158.232044
SDS	150	0.30	AUCUNE	RANDOM	SPTFAM	609.621406	1398.756357	1165.162162
SDS	150	0.30	AUCUNE	RANDOM	SPTFAM	612.4781089	1418.600633	1155.297297
SDS	150	0.30	AUCUNE	RANDOM	SPTFAM	610.071309	1420.475561	1169.23913
SDS	150	0.30	AUCUNE	RANDOM	SPTFAM	580.5442789	1394.770339	1164.281915
SDS	150	0.30	AUCUNE	SPT	RANFAM	792.4064834	1308.087407	1146.961326
SDS	150	0.30	AUCUNE	SPT	RANFAM	750.7346702	1288.27229	1156
SDS	150	0.30	AUCUNE	SPT	RANFAM	768.8024548	1291.09786	1146
SDS	150	0.30	AUCUNE	SPT	RANFAM	762.7626355	1305.2402	1160.679348
SDS	150	0.30	AUCUNE	SPT	RANFAM	725.9541748	1273.691129	1154.126984
SDS	150	0.30	AUCUNE	SPT	SPTFAM	820.131561	1277.402863	1158.232044
SDS	150	0.30	AUCUNE	SPT	SPTFAM	774.8547236	1260.10875	1165.162162
SDS	150	0.30	AUCUNE	SPT	SPTFAM	794.8467809	1261.483114	1155.297297
SDS	150	0.30	AUCUNE	SPT	SPTFAM	789.836093	1275.522663	1169.23913
SDS	150	0.30	AUCUNE	SPT	SPTFAM	747.1329404	1251.044696	1164.281915
SDS	150	0.30	RED20%	RANDOM	RANFAM	603.0151286	1430.200706	1124.20765
SDS	150	0.30	RED20%	RANDOM	RANFAM	589.0970125	1378.702232	1129.521277
SDS	150	0.30	RED20%	RANDOM	RANFAM	587.9532818	1401.890278	1124.358289
SDS	150	0.30	RED20%	RANDOM	RANFAM	587.091781	1411.052092	1134.462366
SDS	150	0.30	RED20%	RANDOM	RANFAM	557.7273461	1370.544339	1127.382199
SDS	150	0.30	RED20%	RANDOM	SPTFAM	633.9945955	1399.755953	1134.725275
SDS	150	0.30	RED20%	RANDOM	SPTFAM	614.8668644	1352.391004	1138.850267
SDS	150	0.30	RED20%	RANDOM	SPTFAM	613.2582563	1378.683785	1133.475936
SDS	150	0.30	RED20%	RANDOM	SPTFAM	615.9699535	1380.306275	1142.972973
SDS	150	0.30	RED20%	RANDOM	SPTFAM	579.9888132	1348.538262	1138.078947
SDS	150	0.30	RED20%	SPT	RANFAM	781.8633843	1331.284994	1124.20765
SDS	150	0.30	RED20%	SPT	RANFAM	739.9415723	1304.208108	1129.521277
SDS	150	0.30	RED20%	SPT	RANFAM	763.2790468	1303.595399	1124.358289
SDS	150	0.30	RED20%	SPT	RANFAM	755.7317235	1325.321535	1134.462366
SDS	150	0.30	RED20%	SPT	RANFAM	716.2960749	1295.961557	1127.382199
SDS	150	0.30	RED20%	SPT	SPTFAM	812.7931442	1298.279804	1134.725275
SDS	150	0.30	RED20%	SPT	SPTFAM	765.324004	1276.759631	1138.850267
SDS	150	0.30	RED20%	SPT	SPTFAM	788.6869221	1276.967966	1133.475936
SDS	150	0.30	RED20%	SPT	SPTFAM	784.2742497	1295.0194	1142.972973
SDS	150	0.30	RED20%	SPT	SPTFAM	738.6768757	1271.97942	1138.078947
SDS	300	0.10	AUCUNE	RANDOM	RANFAM	672.2528009	1104.318587	743.3962264
SDS	300	0.10	AUCUNE	RANDOM	RANFAM	648.2268415	1067.623186	750.437788
SDS	300	0.10	AUCUNE	RANDOM	RANFAM	651.952028	1086.753706	744.2165899
SDS	300	0.10	AUCUNE	RANDOM	RANFAM	644.5395126	1101.36315	755.9490741
SDS	300	0.10	AUCUNE	RANDOM	RANFAM	638.424343	1070.509544	751.6666667
SDS	300	0.10	AUCUNE	RANDOM	SPTFAM	737.922093	1045.660901	770.1435407
SDS	300	0.10	AUCUNE	RANDOM	SPTFAM	707.462775	1007.944444	774.8372093
SDS	300	0.10	AUCUNE	RANDOM	SPTFAM	715.0203649	1021.559044	770.1162791
SDS	300	0.10	AUCUNE	RANDOM	SPTFAM	706.6148918	1033.040006	778.9953271
SDS	300	0.10	AUCUNE	RANDOM	SPTFAM	693.6382682	1016.22486	777.8571429
SDS	300	0.10	AUCUNE	SPT	RANFAM	733.741523	1185.800144	743.3962264
SDS	300	0.10	AUCUNE	SPT	RANFAM	704.7425384	1148.453696	750.437788
SDS	300	0.10	AUCUNE	SPT	RANFAM	719.6238802	1150.75266	744.2165899
SDS	300	0.10	AUCUNE	SPT	RANFAM	709.0384561	1173.60741	755.9490741
SDS	300	0.10	AUCUNE	SPT	RANFAM	684.0331813	1155.938338	751.6666667
SDS	300	0.10	AUCUNE	SPT	SPTFAM	799.4459951	1127.525658	770.1435407
SDS	300	0.10	AUCUNE	SPT	SPTFAM	764.2735609	1088.252257	774.8372093
SDS	300	0.10	AUCUNE	SPT	SPTFAM	782.4914342	1084.503752	770.1162791
SDS	300	0.10	AUCUNE	SPT	SPTFAM	769.7394767	1108.427901	778.9953271
SDS	300	0.10	AUCUNE	SPT	SPTFAM	739.3022086	1102.345261	777.8571429
SDS	300	0.10	RED20%	RANDOM	RANFAM	671.8696774	1104.999713	738.7264151

SDS	300	0.10	RED20%	RANDOM	RANFAM	653.9807826	1060.907812	747.2706422
SDS	300	0.10	RED20%	RANDOM	RANFAM	651.8730603	1088.135887	740.9907834
SDS	300	0.10	RED20%	RANDOM	RANFAM	648.7930259	1095.355174	751.4583333
SDS	300	0.10	RED20%	RANDOM	RANFAM	637.1613438	1070.720725	749.9315068
SDS	300	0.10	RED20%	RANDOM	SPTFAM	741.0267131	1040.645616	765.6666667
SDS	300	0.10	RED20%	RANDOM	SPTFAM	713.0438213	1000.300268	771.5348837
SDS	300	0.10	RED20%	RANDOM	SPTFAM	716.9970335	1020.025427	766.3488372
SDS	300	0.10	RED20%	RANDOM	SPTFAM	709.0501563	1029.641802	773.8551402
SDS	300	0.10	RED20%	RANDOM	SPTFAM	693.019263	1015.224316	776.5668203
SDS	300	0.10	RED20%	SPT	RANFAM	732.3936658	1173.844941	738.7264151
SDS	300	0.10	RED20%	SPT	RANFAM	699.8903594	1145.951797	747.2706422
SDS	300	0.10	RED20%	SPT	RANFAM	714.0513	1163.936679	740.9907834
SDS	300	0.10	RED20%	SPT	RANFAM	705.5476659	1177.183071	751.4583333
SDS	300	0.10	RED20%	SPT	RANFAM	683.1555893	1153.977339	749.9315068
SDS	300	0.10	RED20%	SPT	SPTFAM	800.6891696	1110.520616	765.6666667
SDS	300	0.10	RED20%	SPT	SPTFAM	759.0209948	1085.997605	771.5348837
SDS	300	0.10	RED20%	SPT	SPTFAM	779.0540158	1095.362698	766.3488372
SDS	300	0.10	RED20%	SPT	SPTFAM	764.7957447	1116.381135	773.8551402
SDS	300	0.10	RED20%	SPT	SPTFAM	739.0986741	1099.181577	776.5668203
SDS	300	0.30	AUCUNE	RANDOM	RANFAM	644.4935815	1245.780327	998.9322917
SDS	300	0.30	AUCUNE	RANDOM	RANFAM	618.6382979	1197.860693	993.3585859
SDS	300	0.30	AUCUNE	RANDOM	RANFAM	613.506337	1232.679862	994.9238579
SDS	300	0.30	AUCUNE	RANDOM	RANFAM	627.19136	1204.97645	987.5380711
SDS	300	0.30	AUCUNE	RANDOM	RANFAM	609.1987739	1193.052107	994.85
SDS	300	0.30	AUCUNE	RANDOM	SPTFAM	701.6522581	1190.780645	1013.717278
SDS	300	0.30	AUCUNE	RANDOM	SPTFAM	667.4203125	1154.075938	1007.918782
SDS	300	0.30	AUCUNE	RANDOM	SPTFAM	671.0050275	1175.945326	1007.933674
SDS	300	0.30	AUCUNE	RANDOM	SPTFAM	684.5505653	1151.899026	1001.607143
SDS	300	0.30	AUCUNE	RANDOM	SPTFAM	656.3072178	1152.144355	1006.742424
SDS	300	0.30	AUCUNE	SPT	RANFAM	728.9180591	1401.920791	998.9322917
SDS	300	0.30	AUCUNE	SPT	RANFAM	699.2187354	1349.252913	993.3585859
SDS	300	0.30	AUCUNE	SPT	RANFAM	708.2633391	1370.541073	994.9238579
SDS	300	0.30	AUCUNE	SPT	RANFAM	705.6166094	1357.327349	987.5380711
SDS	300	0.30	AUCUNE	SPT	RANFAM	676.7528366	1355.480681	994.85
SDS	300	0.30	AUCUNE	SPT	SPTFAM	785.9714286	1339.975626	1013.717278
SDS	300	0.30	AUCUNE	SPT	SPTFAM	747.8886804	1302.427767	1007.918782
SDS	300	0.30	AUCUNE	SPT	SPTFAM	764.2724697	1309.802928	1007.933674
SDS	300	0.30	AUCUNE	SPT	SPTFAM	763.5014935	1298.422103	1001.607143
SDS	300	0.30	AUCUNE	SPT	SPTFAM	723.4921345	1311.23211	1006.742424
SDS	300	0.30	RED20%	RANDOM	RANFAM	650.8306567	1205.170138	976.7010309
SDS	300	0.30	RED20%	RANDOM	RANFAM	641.8921207	1146.927516	975.8542714
SDS	300	0.30	RED20%	RANDOM	RANFAM	627.0627512	1194.701082	970.7070707
SDS	300	0.30	RED20%	RANDOM	RANFAM	631.0945359	1181.668892	978.030303
SDS	300	0.30	RED20%	RANDOM	RANFAM	613.3149666	1161	976.1442786
SDS	300	0.30	RED20%	RANDOM	SPTFAM	710.6172563	1155.013447	992.9166667
SDS	300	0.30	RED20%	RANDOM	SPTFAM	691.716677	1098.32734	988.8888889
SDS	300	0.30	RED20%	RANDOM	SPTFAM	683.1731339	1141.754558	983.9086294
SDS	300	0.30	RED20%	RANDOM	SPTFAM	686.3669166	1130.955483	991.7258883
SDS	300	0.30	RED20%	RANDOM	SPTFAM	661.9845637	1115.999847	989.35
SDS	300	0.30	RED20%	SPT	RANFAM	722.645777	1392.114999	976.7010309
SDS	300	0.30	RED20%	SPT	RANFAM	688.2005232	1361.298553	975.8542714
SDS	300	0.30	RED20%	SPT	RANFAM	709.2725023	1359.817971	970.7070707
SDS	300	0.30	RED20%	SPT	RANFAM	703.7371446	1361.493864	978.030303
SDS	300	0.30	RED20%	SPT	RANFAM	677.2949848	1350.143921	976.1442786
SDS	300	0.30	RED20%	SPT	SPTFAM	781.9566192	1338.963663	992.9166667
SDS	300	0.30	RED20%	SPT	SPTFAM	736.8202386	1310.816519	988.8888889
SDS	300	0.30	RED20%	SPT	SPTFAM	765.3366568	1302.15423	983.9086294
SDS	300	0.30	RED20%	SPT	SPTFAM	759.586056	1305.085665	991.7258883
SDS	300	0.30	RED20%	SPT	SPTFAM	725.8138005	1299.591493	989.35
GSU	150	0.10	AUCUNE	RANDOM	RANFAM	638.4314178	1320.075513	755.7582938
GSU	150	0.10	AUCUNE	RANDOM	RANFAM	616.7467133	1271.28042	759.1935484
GSU	150	0.10	AUCUNE	RANDOM	RANFAM	627.2597805	1281.564171	755.3472222
GSU	150	0.10	AUCUNE	RANDOM	RANFAM	614.8420384	1303.397798	760.4883721
GSU	150	0.10	AUCUNE	RANDOM	RANFAM	603.4697791	1267.076143	761.1926606

GSU	150	0.10	AUCUNE	RANDOM	SPTFAM	657.4489796	1297.54552	759.1428571
GSU	150	0.10	AUCUNE	RANDOM	SPTFAM	632.2770848	1251.214576	761.4583333
GSU	150	0.10	AUCUNE	RANDOM	SPTFAM	643.5468508	1261.602931	758.4883721
GSU	150	0.10	AUCUNE	RANDOM	SPTFAM	632.5549272	1280.632405	763.7674419
GSU	150	0.10	AUCUNE	RANDOM	SPTFAM	618.5371246	1247.902948	765.412844
GSU	150	0.10	AUCUNE	SPT	RANFAM	827.8839659	1002.207967	755.7582938
GSU	150	0.10	AUCUNE	SPT	RANFAM	783.2266816	985.8533072	759.1935484
GSU	150	0.10	AUCUNE	SPT	RANFAM	799.9292943	1000.807703	755.3472222
GSU	150	0.10	AUCUNE	SPT	RANFAM	795.4763047	1001.572497	760.4883721
GSU	150	0.10	AUCUNE	SPT	RANFAM	766.9282343	991.5009717	761.1926606
GSU	150	0.10	AUCUNE	SPT	SPTFAM	846.5485851	980.8302978	759.1428571
GSU	150	0.10	AUCUNE	SPT	SPTFAM	798.5398404	967.4825655	761.4583333
GSU	150	0.10	AUCUNE	SPT	SPTFAM	815.8048471	981.9819642	758.4883721
GSU	150	0.10	AUCUNE	SPT	SPTFAM	813.4	979.0607345	763.7674419
GSU	150	0.10	AUCUNE	SPT	SPTFAM	782.0655738	973.0295916	765.412844
GSU	150	0.10	RED20%	RANDOM	RANFAM	640.8226644	1307.077134	752.5829384
GSU	150	0.10	RED20%	RANDOM	RANFAM	613.8890907	1271.750943	756.0599078
GSU	150	0.10	RED20%	RANDOM	RANFAM	628.3637386	1274.98707	753.1712963
GSU	150	0.10	RED20%	RANDOM	RANFAM	616.7651708	1295.159046	757.7906977
GSU	150	0.10	RED20%	RANDOM	RANFAM	603.7103984	1262.073025	758.440367
GSU	150	0.10	RED20%	RANDOM	SPTFAM	659.6403293	1285.618807	756.9194313
GSU	150	0.10	RED20%	RANDOM	SPTFAM	629.709709	1251.696838	759.7465438
GSU	150	0.10	RED20%	RANDOM	SPTFAM	645.4696607	1254.44094	756.8981481
GSU	150	0.10	RED20%	RANDOM	SPTFAM	634.7135799	1272.966262	761.0697674
GSU	150	0.10	RED20%	RANDOM	SPTFAM	619.3842412	1241.698999	762.6605505
GSU	150	0.10	RED20%	SPT	RANFAM	824.5407781	1004.649424	752.5829384
GSU	150	0.10	RED20%	SPT	RANFAM	779.8723642	987.014942	756.0599078
GSU	150	0.10	RED20%	SPT	RANFAM	796.3932584	1000.369242	753.1712963
GSU	150	0.10	RED20%	SPT	RANFAM	793.4657457	1002.60812	757.7906977
GSU	150	0.10	RED20%	SPT	RANFAM	763.9465723	995.0745212	758.440367
GSU	150	0.10	RED20%	SPT	SPTFAM	843.7106746	983.3440705	756.9194313
GSU	150	0.10	RED20%	SPT	SPTFAM	796.0797314	967.5030074	759.7465438
GSU	150	0.10	RED20%	SPT	SPTFAM	813.3912187	980.3622291	756.8981481
GSU	150	0.10	RED20%	SPT	SPTFAM	811.5754797	980.2793454	761.0697674
GSU	150	0.10	RED20%	SPT	SPTFAM	779.4495349	976.4692489	762.6605505
GSU	150	0.30	AUCUNE	RANDOM	RANFAM	655.7296345	1356.020549	796.884058
GSU	150	0.30	AUCUNE	RANDOM	RANFAM	617.1612949	1319.552482	795.6308411
GSU	150	0.30	AUCUNE	RANDOM	RANFAM	623.8354575	1340.007017	793.685446
GSU	150	0.30	AUCUNE	RANDOM	RANFAM	623.07806	1337.371215	799.4575472
GSU	150	0.30	AUCUNE	RANDOM	RANFAM	605.7575458	1314.993089	798.5348837
GSU	150	0.30	AUCUNE	RANDOM	SPTFAM	669.4776163	1336.768237	796.9565217
GSU	150	0.30	AUCUNE	RANDOM	SPTFAM	630.0520314	1300.802709	794.9065421
GSU	150	0.30	AUCUNE	RANDOM	SPTFAM	637.3740513	1320.885723	793.0516432
GSU	150	0.30	AUCUNE	RANDOM	SPTFAM	636.9367379	1317.766318	799.2688679
GSU	150	0.30	AUCUNE	RANDOM	SPTFAM	617.4479549	1297.947109	798.4651163
GSU	150	0.30	AUCUNE	SPT	RANFAM	840.8492226	1010.519947	796.884058
GSU	150	0.30	AUCUNE	SPT	RANFAM	792.7831789	985.9823236	795.6308411
GSU	150	0.30	AUCUNE	SPT	RANFAM	807.086639	1002.434341	793.685446
GSU	150	0.30	AUCUNE	SPT	RANFAM	802.776901	1007.251363	799.4575472
GSU	150	0.30	AUCUNE	SPT	RANFAM	774.5961376	994.1960812	798.5348837
GSU	150	0.30	AUCUNE	SPT	SPTFAM	854.5897624	992.5825755	796.9565217
GSU	150	0.30	AUCUNE	SPT	SPTFAM	805.5951838	969.0129667	794.9065421
GSU	150	0.30	AUCUNE	SPT	SPTFAM	820.6198655	984.5384285	793.0516432
GSU	150	0.30	AUCUNE	SPT	SPTFAM	816.5383733	989.2308134	799.2688679
GSU	150	0.30	AUCUNE	SPT	SPTFAM	786.0772266	979.7512683	798.4651163
GSU	150	0.30	RED20%	RANDOM	RANFAM	652.9026107	1361.924465	795.3864734
GSU	150	0.30	RED20%	RANDOM	RANFAM	615.0292314	1330.066448	794.8831776
GSU	150	0.30	RED20%	RANDOM	RANFAM	621.3706143	1347.070743	792.6056338
GSU	150	0.30	RED20%	RANDOM	RANFAM	620.1414634	1343.494261	799.009434
GSU	150	0.30	RED20%	RANDOM	RANFAM	602.5500423	1324.024105	796.5813953
GSU	150	0.30	RED20%	RANDOM	SPTFAM	666.9599707	1342.329326	795.3140097
GSU	150	0.30	RED20%	RANDOM	SPTFAM	627.4263511	1311.61443	794.3925234
GSU	150	0.30	RED20%	RANDOM	SPTFAM	634.829373	1327.09791	792.1596244
GSU	150	0.30	RED20%	RANDOM	SPTFAM	633.4924688	1324.715105	799.1273585

GSU	150	0.30	RED20%	RANDOM	SPTFAM	614.1358895	1307.063011	796.6511628
GSU	150	0.30	RED20%	SPT	RANFAM	840.7194603	1009.972137	795.4347826
GSU	150	0.30	RED20%	SPT	RANFAM	793.2034768	984.7601881	794.8831776
GSU	150	0.30	RED20%	SPT	RANFAM	808.8320458	1000.312876	792.6056338
GSU	150	0.30	RED20%	SPT	RANFAM	803.4828576	1005.982786	799.009434
GSU	150	0.30	RED20%	SPT	RANFAM	774.447717	993.5693348	796.5813953
GSU	150	0.30	RED20%	SPT	SPTFAM	854.5412817	991.6896906	795.3140097
GSU	150	0.30	RED20%	SPT	SPTFAM	805.8946169	967.7445172	794.3925234
GSU	150	0.30	RED20%	SPT	SPTFAM	822.2191076	982.354119	792.1596244
GSU	150	0.30	RED20%	SPT	SPTFAM	816.7776503	988.9000143	799.1273585
GSU	150	0.30	RED20%	SPT	SPTFAM	785.8461322	978.8812174	796.6511628
GSU	300	0.10	AUCUNE	RANDOM	RANFAM	632.4802051	1268.669467	723.691588
GSU	300	0.10	AUCUNE	RANDOM	RANFAM	605.8215716	1238.071953	727.739726
GSU	300	0.10	AUCUNE	RANDOM	RANFAM	611.4258127	1253.44554	725.3440367
GSU	300	0.10	AUCUNE	RANDOM	RANFAM	609.2920156	1257.116555	730.8986175
GSU	300	0.10	AUCUNE	RANDOM	RANFAM	595.219105	1230.347338	728.7782805
GSU	300	0.10	AUCUNE	RANDOM	SPTFAM	663.8742129	1237.852175	734.9765258
GSU	300	0.10	AUCUNE	RANDOM	SPTFAM	633.3483817	1209.190721	738.8302752
GSU	300	0.10	AUCUNE	RANDOM	SPTFAM	643.6169738	1223.093104	737.281106
GSU	300	0.10	AUCUNE	RANDOM	SPTFAM	640.3913957	1226.753784	741.1111111
GSU	300	0.10	AUCUNE	RANDOM	SPTFAM	621.6662996	1202.531801	739.8181818
GSU	300	0.10	AUCUNE	SPT	RANFAM	777.1493664	1053.20561	723.691588
GSU	300	0.10	AUCUNE	SPT	RANFAM	741.5804312	1029.726368	727.739726
GSU	300	0.10	AUCUNE	SPT	RANFAM	752.8546617	1042.463665	725.3440367
GSU	300	0.10	AUCUNE	SPT	RANFAM	761.9025105	1037.528731	730.8986175
GSU	300	0.10	AUCUNE	SPT	RANFAM	726.0253182	1032.760914	728.7782805
GSU	300	0.10	AUCUNE	SPT	SPTFAM	808.624857	1017.553919	734.9765258
GSU	300	0.10	AUCUNE	SPT	SPTFAM	769.1286607	998.1576683	738.8302752
GSU	300	0.10	AUCUNE	SPT	SPTFAM	784.489608	1006.709165	737.281106
GSU	300	0.10	AUCUNE	SPT	SPTFAM	792.4857863	1004.07604	741.1111111
GSU	300	0.10	AUCUNE	SPT	SPTFAM	752.1677437	1001.81342	739.8181818
GSU	300	0.10	RED20%	RANDOM	RANFAM	616.0718051	1290.966235	723.317757
GSU	300	0.10	RED20%	RANDOM	RANFAM	611.0196106	1230.39663	727.4657534
GSU	300	0.10	RED20%	RANDOM	RANFAM	605.6279845	1261.444892	723.4633028
GSU	300	0.10	RED20%	RANDOM	RANFAM	606.2020357	1259.303542	729.0825688
GSU	300	0.10	RED20%	RANDOM	RANFAM	589.8071448	1239.558582	727.3303167
GSU	300	0.10	RED20%	RANDOM	SPTFAM	649.5823673	1258.804065	735.3051643
GSU	300	0.10	RED20%	RANDOM	SPTFAM	637.3722577	1203.185921	738.3256881
GSU	300	0.10	RED20%	RANDOM	SPTFAM	638.8991069	1231.288027	735.1612903
GSU	300	0.10	RED20%	RANDOM	SPTFAM	638.5966704	1227.371992	739.4009217
GSU	300	0.10	RED20%	RANDOM	SPTFAM	617.2125757	1211.319758	739.0454545
GSU	300	0.10	RED20%	SPT	RANFAM	774.1378967	1059.661164	723.317757
GSU	300	0.10	RED20%	SPT	RANFAM	735.3866243	1040.235042	727.4657534
GSU	300	0.10	RED20%	SPT	RANFAM	749.8116324	1046.764853	723.4633028
GSU	300	0.10	RED20%	SPT	RANFAM	751.6158673	1052.399052	729.0825688
GSU	300	0.10	RED20%	SPT	RANFAM	719.4082638	1043.252702	727.3303167
GSU	300	0.10	RED20%	SPT	SPTFAM	807.1809469	1022.61808	735.3051643
GSU	300	0.10	RED20%	SPT	SPTFAM	762.092423	1009.786428	738.3256881
GSU	300	0.10	RED20%	SPT	SPTFAM	782.8569038	1011.129847	735.1612903
GSU	300	0.10	RED20%	SPT	SPTFAM	783.4501329	1016.736047	739.4009217
GSU	300	0.10	RED20%	SPT	SPTFAM	746.4869434	1011.43994	739.0454545
GSU	300	0.30	AUCUNE	RANDOM	RANFAM	642.8385157	1342.956146	792.3429952
GSU	300	0.30	AUCUNE	RANDOM	RANFAM	609.694049	1307.574744	791.6588785
GSU	300	0.30	AUCUNE	RANDOM	RANFAM	620.7845296	1316.733486	789.882691
GSU	300	0.30	AUCUNE	RANDOM	RANFAM	614.615142	1326.517924	796.6037736
GSU	300	0.30	AUCUNE	RANDOM	RANFAM	596.0599916	1305.522743	796.5740741
GSU	300	0.30	AUCUNE	RANDOM	SPTFAM	655.608868	1326.591543	794.9758454
GSU	300	0.30	AUCUNE	RANDOM	SPTFAM	621.1796407	1293.642715	794.6261682
GSU	300	0.30	AUCUNE	RANDOM	SPTFAM	634.5258337	1301.928725	792.5821596
GSU	300	0.30	AUCUNE	RANDOM	SPTFAM	628.3793054	1309.557549	799.4103774
GSU	300	0.30	AUCUNE	RANDOM	SPTFAM	607.6608426	1291.227138	797.8604651
GSU	300	0.30	AUCUNE	SPT	RANFAM	835.4743552	1009.321366	792.3429952
GSU	300	0.30	AUCUNE	SPT	RANFAM	790.8784648	982.4601279	791.6588785
GSU	300	0.30	AUCUNE	SPT	RANFAM	801.656	999.9277143	789.882691

```

GSU 300 0.30 AUCUNE SPT      RANFAM 801.5434471 1002.490536 796.6037736
GSU 300 0.30 AUCUNE SPT      RANFAM 767.9545134 1000.764259 796.5740741
GSU 300 0.30 AUCUNE SPT      SPTFAM 848.682158  991.9901774 794.9758454
GSU 300 0.30 AUCUNE SPT      SPTFAM 802.2734653 967.1679248 794.6261682
GSU 300 0.30 AUCUNE SPT      SPTFAM 815.5082976 983.954364 792.5821596
GSU 300 0.30 AUCUNE SPT      SPTFAM 815.1889527 985.2519369 799.4103774
GSU 300 0.30 AUCUNE SPT      SPTFAM 779.6491475 985.6422432 797.8604651
GSU 300 0.30 RED20% RANDOM  RANFAM 623.2564253 1370.751799 791.2560386
GSU 300 0.30 RED20% RANDOM  RANFAM 603.4479863 1318.902377 790.4906542
GSU 300 0.30 RED20% RANDOM  RANFAM 610.1473699 1332.510435 788.42723
GSU 300 0.30 RED20% RANDOM  RANFAM 606.857409  1336.8362  795.7075472
GSU 300 0.30 RED20% RANDOM  RANFAM 610.1473699 1332.510435 795.2314815
GSU 300 0.30 RED20% RANDOM  SPTFAM 637.0576555 1352.922783 794.1545894
GSU 300 0.30 RED20% RANDOM  SPTFAM 614.5227985 1305.490596 793.364486
GSU 300 0.30 RED20% RANDOM  SPTFAM 624.0998712 1317.230505 790.9389671
GSU 300 0.30 RED20% RANDOM  SPTFAM 620.6748924 1319.593687 798.3254717
GSU 300 0.30 RED20% RANDOM  SPTFAM 594.6323799 1310.07313 796.7906977
GSU 300 0.30 RED20% SPT     RANFAM 831.3735346 1014.890533 791.2560386
GSU 300 0.30 RED20% SPT     RANFAM 787.3334281 987.2673112 790.4906542
GSU 300 0.30 RED20% SPT     RANFAM 799.4517281 1003.394459 788.42723
GSU 300 0.30 RED20% SPT     RANFAM 799.0086059 1006.583907 795.7075472
GSU 300 0.30 RED20% SPT     RANFAM 767.1923185 1002.908554 795.2314815
GSU 300 0.30 RED20% SPT     SPTFAM 845.1628793 996.8051605 794.1545894
GSU 300 0.30 RED20% SPT     SPTFAM 798.7140418 972.6027884 793.364486
GSU 300 0.30 RED20% SPT     SPTFAM 813.0923253 987.1614978 790.9389671
GSU 300 0.30 RED20% SPT     SPTFAM 812.720373 989.4615495 798.3254717
GSU 300 0.30 RED20% SPT     SPTFAM 777.94899925 988.7355221 796.7906977
;
proc anova;
  classes methode feeder rho reduct interf intraf;
  model wip = methode feeder rho reduct interf intraf methode*feeder methode*rho
    methode*reduct methode*interf methode*intraf feeder*rho feeder*reduct
    feeder*interf   feeder*intraf   rho*reduct   rho*interf   rho*intraf
    reduct*interf
    reduct*intraf interf*intraf;
  means methode/duncan alpha=0.05;
  means feeder/duncan alpha=0.05;
  means rho/duncan alpha=0.05;
  means reduct/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;
run;
proc anova;
  classes methode feeder rho reduct interf intraf;
  model tpas = methode feeder rho reduct interf intraf methode*feeder
    methode*rho methode*reduct methode*interf methode*intraf
    feeder*rho feeder*reduct feeder*interf feeder*intraf rho*reduct
    rho*interf
    rho*intraf reduct*interf reduct*intraf interf*intraf;
  means methode/duncan alpha=0.05;
  means feeder/duncan alpha=0.05;
  means rho/duncan alpha=0.05;
  means reduct/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;
run;
proc anova;
  classes methode feeder rho reduct interf intraf;
  model setup = methode feeder rho reduct interf intraf methode*feeder
    methode*rho
    methode*reduct methode*interf methode*intraf
    feeder*rho   feeder*reduct   feeder*interf   feeder*intraf   rho*reduct
    rho*interf
    rho*intraf reduct*interf reduct*intraf interf*intraf;

```

```

means methode/duncan alpha=0.05;
means feeder/duncan alpha=0.05;
means rho/duncan alpha=0.05;
means reduct/duncan alpha=0.05;
means interf/duncan alpha=0.05;
means intraf/duncan alpha=0.05;
run;

```

Les résultats sont les suivants:

36

```

Simulation Ligne SMT

Analysis of Variance Procedure
      Class Level Information

      Class      Levels      Values
      METHODE      2      GSU SDS
      FEEDER      2      150 300
      RHO          2      0.1 0.3
      REDUCT      2      AUCUNE RED20%
      INTERF      2      RANDOM SPT
      INTRAF      2      RANFAM SPTFAM

      Number of observations in data set = 320

```

37

```

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: WIP

      Source      DF      Sum of Squares      Mean Square      F Value      Pr > F
      Model       21      1825585.6125      86932.6482      189.26      0.0001
      Error       298      136883.5009      459.3406
      Corrected Total      319      1962469.1134

      R-Square      C.V.      Root MSE      WIP Mean
      0.930249      3.047973      21.432233      703.16345

```

Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	15966.5367	15966.5367	34.76	0.0001
FEEDER	1	447.5276	447.5276	0.97	0.3244
RHO	1	709.3429	709.3429	1.54	0.2150
REDUCT	1	303.0351	303.0351	0.66	0.4173
INTERF	1	1546783.2722	1546783.2722	3367.40	0.0001
INTRAF	1	71230.8551	71230.8551	155.07	0.0001
METHODE*FEEDER	1	13015.2208	13015.2208	28.33	0.0001
METHODE*RHO	1	16883.9765	16883.9765	36.76	0.0001
METHODE*REDUCT	1	129.8530	129.8530	0.28	0.5953
METHODE*INTERF	1	75546.1045	75546.1045	164.47	0.0001
METHODE*INTRAF	1	11532.2977	11532.2977	25.11	0.0001
FEEDER*RHO	1	443.8241	443.8241	0.97	0.3264
FEEDER*REDUCT	1	0.2584	0.2584	0.00	0.9811
FEEDER*INTERF	1	49215.0691	49215.0691	107.14	0.0001
FEEDER*INTRAF	1	6841.4876	6841.4876	14.89	0.0001
RHO*REDUCT	1	9.9061	9.9061	0.02	0.8833
RHO*INTERF	1	15192.2813	15192.2813	33.07	0.0001
RHO*INTRAF	1	1189.6251	1189.6251	2.59	0.1086
REDUCT*INTERF	1	141.8491	141.8491	0.31	0.5788
REDUCT*INTRAF	1	2.7086	2.7086	0.01	0.9388
INTERF*INTRAF	1	0.5810	0.5810	0.00	0.9717

38

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means	2
Critical Range	4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	710.227	160	GSU
B	696.100	160	SDS

39

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means 2
Critical Range 4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	FEEDER
A	704.346	160	150
A	701.981	160	300

40

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means 2
Critical Range 4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	RHO
A	704.652	160	0.1
A	701.675	160	0.3

41

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means 2
Critical Range 4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	REDUCT
A	704.137	160	AUCUNE
A			
A	702.190	160	RED20%

42

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means 2
Critical Range 4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	772.688	160	SPT
B	633.639	160	RANDOM

43

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 459.3406

Number of Means 2
Critical Range 4.716

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	718.083	160	SPTFAM
B	688.244	160	RANFAM

Simulation Ligne SMT

Analysis of Variance Procedure
Class Level Information

Class	Levels	Values
METHODE	2	GSU SDS
FEEDER	2	150 300
RHO	2	0.1 0.3
REDUCT	2	AUCUNE RED20%
INTERF	2	RANDOM SPT
INTRAF	2	RANFAM SPTFAM

Number of observations in data set = 320

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: TPAS					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	21	5724563.7937	272598.2759	242.38	0.0001
Error	298	335155.5033	1124.6829		
Corrected Total	319	6059719.2970			
R-Square		C.V.	Root MSE	TPAS Mean	
0.944691		2.832439	33.536292	1184.0076	
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	486983.3607	486983.3607	433.00	0.0001
FEEDER	1	192781.9779	192781.9779	171.41	0.0001
RHO	1	672491.9568	672491.9568	597.94	0.0001
REDUCT	1	80.1674	80.1674	0.07	0.7897
INTERF	1	1471582.2274	1471582.2274	1308.44	0.0001
INTRAF	1	81722.7475	81722.7475	72.66	0.0001
METHODE*FEEDER	1	188257.5456	188257.5456	167.39	0.0001
METHODE*RHO	1	340117.1842	340117.1842	302.41	0.0001
METHODE*REDUCT	1	2721.0210	2721.0210	2.42	0.1209
METHODE*INTERF	1	1932058.7455	1932058.7455	1717.87	0.0001

METHODE*INTRAF	1	8589.4489	8589.4489	7.64	0.0061
FEEDER*RHO	1	192.1998	192.1998	0.17	0.6796
FEEDER*REDUCT	1	17.2972	17.2972	0.02	0.9014
FEEDER*INTERF	1	310172.2675	310172.2675	275.79	0.0001
FEEDER*INTRAF	1	5644.6977	5644.6977	5.02	0.0258
RHO*REDUCT	1	888.8240	888.8240	0.79	0.3747
RHO*INTERF	1	27031.5571	27031.5571	24.03	0.0001
RHO*INTRAF	1	1071.0597	1071.0597	0.95	0.3299
REDUCT*INTERF	1	2138.9189	2138.9189	1.90	0.1689
REDUCT*INTRAF	1	5.2438	5.2438	0.00	0.9456
INTERF*INTRAF	1	15.3450	15.3450	0.01	0.9071

46

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means	2
Critical Range	7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	1223.018	160	SDS
B	1144.997	160	GSU

47

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means	2
Critical Range	7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	FEEDER
-----------------	------	---	--------

A	1208.552	160	150
B	1159.463	160	300

48

Simulation Ligne SMT
 Analysis of Variance Procedure
 Duncan's Multiple Range Test for variable: TPAS
 NOTE: This test controls the type I comparisonwise error rate, not
 the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means 2
 Critical Range 7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	RHO
A	1229.850	160	0.3
B	1138.165	160	0.1

49

Simulation Ligne SMT
 Analysis of Variance Procedure
 Duncan's Multiple Range Test for variable: TPAS
 NOTE: This test controls the type I comparisonwise error rate, not
 the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means 2
 Critical Range 7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	REDUCT
A	1184.508	160	AUCUNE
A	1183.507	160	RED20%

50

Simulation Ligne SMT
 Analysis of Variance Procedure
 Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means 2
 Critical Range 7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	1251.821	160	RANDOM
B	1116.194	160	SPT

51

Simulation Ligne SMT
 Analysis of Variance Procedure
 Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 1124.683

Number of Means 2
 Critical Range 7.379

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	1199.988	160	RANFAM
B	1168.027	160	SPTFAM

52

Simulation Ligne SMT
 Analysis of Variance Procedure
 Class Level Information

Class	Levels	Values
-------	--------	--------

METHODE	2	GSU SDS
FEEDER	2	150 300
RHO	2	0.1 0.3
REDUCT	2	AUCUNE RED20%
INTERF	2	RANDOM SPT
INTRAF	2	RANFAM SPTFAM

Number of observations in data set = 320

53

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: SETUP

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	21	6055265.8081	288345.9909	13771.94	0.0001
Error	298	6239.2885	20.9372		
Corrected Total	319	6061505.0966			
		R-Square	C.V.	Root MSE	SETUP Mean
		0.998971	0.528629	4.5757196	865.58309

Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	2903607.5340	2903607.5340	99999.99	0.0001
FEEDER	1	679831.0254	679831.0254	32469.99	0.0001
RHO	1	1400950.5578	1400950.5578	66912.00	0.0001
REDUCT	1	4170.0163	4170.0163	199.17	0.0001
INTERF	1	0.0000	0.0000	0.00	0.9995
INTRAF	1	7948.2948	7948.2948	379.63	0.0001
METHODE*FEEDER	1	485586.7921	485586.7921	23192.53	0.0001
METHODE*RHO	1	548160.8684	548160.8684	26181.18	0.0001
METHODE*REDUCT	1	2662.8050	2662.8050	127.18	0.0001
METHODE*INTERF	1	0.0000	0.0000	0.00	0.9995
METHODE*INTRAF	1	2611.9455	2611.9455	124.75	0.0001
FEEDER*RHO	1	16434.8763	16434.8763	784.96	0.0001
FEEDER*REDUCT	1	114.3795	114.3795	5.46	0.0201
FEEDER*INTERF	1	0.0000	0.0000	0.00	0.9995
FEEDER*INTRAF	1	817.1507	817.1507	39.03	0.0001
RHO*REDUCT	1	1361.0148	1361.0148	65.00	0.0001
RHO*INTERF	1	0.0000	0.0000	0.00	0.9995
RHO*INTRAF	1	1007.5606	1007.5606	48.12	0.0001
REDUCT*INTERF	1	0.0000	0.0000	0.00	0.9995
REDUCT*INTRAF	1	0.9870	0.9870	0.05	0.8283
INTERF*INTRAF	1	0.0000	0.0000	0.00	0.9995

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means	2
Critical Range	1.007

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	960.8394	160	SDS
B	770.3267	160	GSU

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means	2
Critical Range	1.007

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	FEEDER
A	911.6751	160	150
B	819.4911	160	300

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means		2
Critical Range		1.007
Means with the same letter are not significantly different.		
Duncan Grouping	Mean	N RHO
A	931.7493	160 0.3
B	799.4169	160 0.1

57

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means		2
Critical Range		1.007
Means with the same letter are not significantly different.		
Duncan Grouping	Mean	N REDUCT
A	869.1930	160 AUCUNE
B	861.9732	160 RED20%

58

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means 2
 Critical Range 1.007

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	865.5832	160	SPT
A	865.5829	160	RANDOM

59

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 298 MSE= 20.93721

Number of Means 2
 Critical Range 1.007

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	870.5669	160	SPTFAM
B	860.5993	160	RANFAM

ANNEXE D

SAS EXPÉRIENCE 2

Voici le programme SAS pour l'analyse de variance et l'étude de Duncan des variables.

```

data matrox;
  options pagesize=60 linesize=80;
  options nodate;
  title;
  title5 'Simulation Ligne SMT';
input methode$ interf$ intrraf$ tpas wip setup ratio;
cards;
SDS      RND      RND      1337.111165 592.1433606 1063.59375 0.4229988
SDS      RND      RND      1363.889121 636.2800129 1092.243243 0.4178887
SDS      RND      RND      1386.944642 637.8637831 1085.625 0.4155314
SDS      RND      RND      1448.464759 669.617825 1121.497175 0.4106848
SDS      RND      RND      1364.478137 616.1192966 1074.552632 0.423807
SDS      RND      SPT      1254.42821 631.8706665 1035.230769 0.4175948
SDS      RND      SPT      1284.191402 680.7017544 1066.256685 0.4113878
SDS      RND      SPT      1293.444641 696.8206653 1055.752688 0.4059005
SDS      RND      SPT      1335.864717 699.0337404 1090.303867 0.4074961
SDS      RND      SPT      1285.003182 678.1817025 1056.43617 0.4129509
SDS      RND      RND/PCB  1270.450791 595.6078612 971.9095477 0.4022012
SDS      RND      RND/PCB  1278.508558 640.6254433 975.0773196 0.3930025
SDS      RND      RND/PCB  1302.016313 645.6846722 992.8645833 0.396797
SDS      RND      RND/PCB  1334.742035 665.3182802 1007.473118 0.3912948
SDS      RND      RND/PCB  1283.08701 614.0075061 957.4371859 0.4000255
SDS      RND      SPT/PCB  1224.584663 626.3289877 969.95 0.402009
SDS      RND      SPT/PCB  1223.360487 668.8236834 973.1025641 0.39601
SDS      RND      SPT/PCB  1259.336393 702.7312899 991.4021164 0.392426
SDS      RND      SPT/PCB  1289.382063 712.3585851 1003.75 0.3878146
SDS      RND      SPT/PCB  1216.054744 663.6739966 960.8333333 0.3980252
SDS      FSPT     RND      1262.610284 728.7033477 1063.59375 0.4229988
SDS      FSPT     RND      1286.015454 777.9420476 1092.243243 0.4178887
SDS      FSPT     RND      1302.104644 782.4758143 1085.625 0.4155314
SDS      FSPT     RND      1370.105151 811.6979558 1121.497175 0.4106848
SDS      FSPT     RND      1287.324834 760.8132721 1074.552632 0.423807
SDS      FSPT     SPT      1185.039582 767.749525 1035.230769 0.4175948
SDS      FSPT     SPT      1222.884865 817.3833623 1066.256685 0.4113878
SDS      FSPT     SPT      1221.256789 835.6083882 1055.752688 0.4059005
SDS      FSPT     SPT      1275.055056 840.1693419 1090.303867 0.4074961
SDS      FSPT     SPT      1233.796723 814.1052966 1056.43617 0.4129509
SDS      FSPT     RND/PCB  1212.271262 734.2531471 971.9095477 0.4022012
SDS      FSPT     RND/PCB  1230.953424 775.5271437 975.0773196 0.3930025
SDS      FSPT     RND/PCB  1240.540237 783.2041853 992.8645833 0.396797
SDS      FSPT     RND/PCB  1305.494047 804.8510901 1007.473118 0.3912948
SDS      FSPT     RND/PCB  1226.559577 762.282472 957.4371859 0.4000255
SDS      FSPT     SPT/PCB  1166.893695 764.860408 969.95 0.402009
SDS      FSPT     SPT/PCB  1178.04141 801.7276786 973.1025641 0.39601
SDS      FSPT     SPT/PCB  1199.747316 840.8349152 991.4021164 0.392426
SDS      FSPT     SPT/PCB  1247.493818 856.0460166 1003.75 0.3878146
SDS      FSPT     SPT/PCB  1187.767907 795.4385183 960.8333333 0.3980252

```

SDS	MS	RND	1519.000801	598.8647436	1063.59375	0.4229988
SDS	MS	RND	1577.282241	637.2244405	1092.243243	0.4178887
SDS	MS	RND	1595.959787	633.936531	1085.625	0.4155314
SDS	MS	RND	1675.442191	660.1025348	1121.497175	0.4106848
SDS	MS	RND	1554.16038	624.851664	1074.552632	0.423807
SDS	MS	SPT	1443.027083	638.2448527	1035.230769	0.4175948
SDS	MS	SPT	1505.159576	668.5992409	1066.256685	0.4113878
SDS	MS	SPT	1536.537683	672.1624618	1055.752688	0.4059005
SDS	MS	SPT	1576.600386	688.5618068	1090.303867	0.4074961
SDS	MS	SPT	1519.038333	670.1592174	1056.43617	0.4129509
SDS	MS	RND/PCB	1457.171683	610.1980958	971.9095477	0.4022012
SDS	MS	RND/PCB	1501.121507	633.9813185	975.0773196	0.3930025
SDS	MS	RND/PCB	1553.335129	624.4879212	992.8645833	0.396797
SDS	MS	RND/PCB	1602.376702	651.8896968	1007.473118	0.3912948
SDS	MS	RND/PCB	1487.82104	626.8954148	957.4371859	0.4000255
SDS	MS	SPT/PCB	1411.963946	642.2379564	969.95	0.402009
SDS	MS	SPT/PCB	1455.330973	658.1736453	973.1025641	0.39601
SDS	MS	SPT/PCB	1508.021644	679.414824	991.4021164	0.392426
SDS	MS	SPT/PCB	1541.668493	705.8142119	1003.75	0.3878146
SDS	MS	SPT/PCB	1471.371596	646.4477612	960.8333333	0.3980252
SDS	FSPTS	RND	1314.932692	708.4009615	1063.59375	0.4229988
SDS	FSPTS	RND	1345.15792	756.8451384	1092.243243	0.4178887
SDS	FSPTS	RND	1351.542567	765.4237343	1085.625	0.4155314
SDS	FSPTS	RND	1434.798038	787.3919869	1121.497175	0.4106848
SDS	FSPTS	RND	1341.081115	741.3986058	1074.552632	0.423807
SDS	FSPTS	SPT	1233.564915	750.1198543	1035.230769	0.4175948
SDS	FSPTS	SPT	1264.504349	802.337498	1066.256685	0.4113878
SDS	FSPTS	SPT	1266.78933	819.8939418	1055.752688	0.4059005
SDS	FSPTS	SPT	1335.843077	818.8647607	1090.303867	0.4074961
SDS	FSPTS	SPT	1286.202799	796.471449	1056.43617	0.4129509
SDS	FSPTS	RND/PCB	1250.756524	721.530089	971.9095477	0.4022012
SDS	FSPTS	RND/PCB	1261.388803	765.0198951	975.0773196	0.3930025
SDS	FSPTS	RND/PCB	1282.148177	769.9346053	992.8645833	0.396797
SDS	FSPTS	RND/PCB	1347.472772	789.9221844	1007.473118	0.3912948
SDS	FSPTS	RND/PCB	1265.560957	749.0271431	957.4371859	0.4000255
SDS	FSPTS	SPT/PCB	1203.255714	752.7116122	969.95	0.402009
SDS	FSPTS	SPT/PCB	1218.1875	787.1830357	973.1025641	0.39601
SDS	FSPTS	SPT/PCB	1232.354196	830.9532929	991.4021164	0.392426
SDS	FSPTS	SPT/PCB	1293.200188	841.062138	1003.75	0.3878146
SDS	FSPTS	SPT/PCB	1226.604058	783.6134338	960.8333333	0.3980252
GSU	RND	RND	1311.934271	600.2342135	817.6179245	0.3609989
GSU	RND	RND	1353.956235	632.2700527	816.884058	0.3512331
GSU	RND	RND	1375.314899	662.827229	834.5792079	0.3543127
GSU	RND	RND	1369.263474	645.9787639	834.8756219	0.3486347
GSU	RND	RND	1379.462274	654.5723389	833.1127451	0.3573072
GSU	RND	SPT	1274.793497	619.286081	803.7149533	0.3575622
GSU	RND	SPT	1303.676571	670.2388766	805.7038835	0.3450025
GSU	RND	SPT	1343.566101	695.1314712	822.7970297	0.3469566
GSU	RND	SPT	1318.850479	677.4099099	814.2647059	0.3467792
GSU	RND	SPT	1339.106733	684.1204013	814.3349754	0.3468171
GSU	RND	RND/PCB	1284.522999	608.4785483	770.1388889	0.3473438
GSU	RND	RND/PCB	1286.213159	632.621998	764.4859813	0.3416333
GSU	RND	RND/PCB	1367.33056	665.5239598	783.3823529	0.3401258
GSU	RND	RND/PCB	1327.053946	644.5164993	772.705314	0.3362336
GSU	RND	RND/PCB	1321.596449	647.7393265	767.3474178	0.3441099
GSU	RND	SPT/PCB	1264.849346	622.6833591	769.8611111	0.3472745
GSU	RND	SPT/PCB	1263.624201	649.0802443	763.5046729	0.3413855
GSU	RND	SPT/PCB	1342.10284	682.9335915	783.0731707	0.3401655
GSU	RND	SPT/PCB	1318.286125	656.9029475	769.8309179	0.3334803
GSU	RND	SPT/PCB	1314.887428	679.8188375	767.5829384	0.3429601
GSU	FSPT	RND	1024.525516	773.4426606	817.6179245	0.3609989
GSU	FSPT	RND	1038.910061	820.9148874	816.884058	0.3512331
GSU	FSPT	RND	1060.742307	842.3857372	834.5792079	0.3543127
GSU	FSPT	RND	1067.915703	831.1963998	834.8756219	0.3486347

```

GSU  FSPT  RND    1079.451202  833.948161  833.1127451  0.3573072
GSU  FSPT  SPT    988.3091505  793.2498575  803.7149533  0.3575622
GSU  FSPT  SPT    1004.02215   850.7272593  805.7038835  0.3450025
GSU  FSPT  SPT    1037.299357   874.865019   822.7970297  0.3469566
GSU  FSPT  SPT    1022.653986   858.0081313  814.2647059  0.3467792
GSU  FSPT  SPT    1041.742512   862.1999128  814.3349754  0.3468171
GSU  FSPT  RND/PCB 998.6293903   786.620118   770.1388889  0.3473438
GSU  FSPT  RND/PCB 1003.335892   807.5034826  764.4859813  0.3416333
GSU  FSPT  RND/PCB 1054.185626   853.9368814  783.3823529  0.3401258
GSU  FSPT  RND/PCB 1035.276758   829.2055954  772.705314  0.3362336
GSU  FSPT  RND/PCB 1029.172467   827.6209666  767.3474178  0.3441099
GSU  FSPT  SPT/PCB 979.2659455   800.8123068  769.8611111  0.3472745
GSU  FSPT  SPT/PCB 980.9828076   823.8020745  763.5046729  0.3413855
GSU  FSPT  SPT/PCB 1029.269937   871.3709122  783.0731707  0.3401655
GSU  FSPT  SPT/PCB 1006.264482   855.051459   769.8309179  0.3334803
GSU  FSPT  SPT/PCB 1008.804271   872.7134776  767.5829384  0.3429601
GSU  MS    RND    1346.572781   598.0068837  817.6179245  0.3609989
GSU  MS    RND    1391.346126   626.0263659  816.884058  0.3512331
GSU  MS    RND    1436.150737   633.2018094  834.5792079  0.3543127
GSU  MS    RND    1420.165323   635.8724901  834.8756219  0.3486347
GSU  MS    RND    1421.099797   644.7668019  833.1127451  0.3573072
GSU  MS    SPT    1311.708256   617.9108798  803.7149533  0.3575622
GSU  MS    SPT    1343.466519   662.3268293  805.7038835  0.3450025
GSU  MS    SPT    1405.468412   669.791167   822.7970297  0.3469566
GSU  MS    SPT    1395.227808   653.0031963  814.2647059  0.3467792
GSU  MS    SPT    1389.379095   668.7676518  814.3349754  0.3468171
GSU  MS    RND/PCB 1323.51351   609.9078244  770.1388889  0.3473438
GSU  MS    RND/PCB 1340.49339   622.4380952  764.4859813  0.3416333
GSU  MS    RND/PCB 1406.581432   660.0612713  783.3823529  0.3401258
GSU  MS    RND/PCB 1387.536088   640.2390587  772.705314  0.3362336
GSU  MS    RND/PCB 1383.343525   635.6318163  767.3474178  0.3441099
GSU  MS    SPT/PCB 1306.650063   622.6495005  769.8611111  0.3472745
GSU  MS    SPT/PCB 1318.894131   637.9616314  763.5046729  0.3413855
GSU  MS    SPT/PCB 1383.912364   676.2621916  783.0731707  0.3401655
GSU  MS    SPT/PCB 1379.163262   648.1411105  769.8309179  0.3334803
GSU  MS    SPT/PCB 1363.259794   671.5867628  767.5829384  0.3429601
GSU  FSPTS  RND    1062.570528   754.516055  817.6179245  0.3609989
GSU  FSPTS  RND    1084.17666   797.7570927  816.884058  0.3512331
GSU  FSPTS  RND    1106.49001   817.8263089  834.5792079  0.3543127
GSU  FSPTS  RND    1114.480755   807.2122055  834.8756219  0.3486347
GSU  FSPTS  RND    1127.258905   809.783956  833.1127451  0.3573072
GSU  FSPTS  SPT    1021.911488   776.5571551  803.7149533  0.3575622
GSU  FSPTS  SPT    1045.48036   829.4923213  805.7038835  0.3450025
GSU  FSPTS  SPT    1079.562153   853.3416204  822.7970297  0.3469566
GSU  FSPTS  SPT    1068.164246   835.997386  814.2647059  0.3467792
GSU  FSPTS  SPT    1080.976011   843.2816226  814.5073892  0.3468669
GSU  FSPTS  RND/PCB 1027.397189   772.5137034  770.1388889  0.3473438
GSU  FSPTS  RND/PCB 1038.866951   791.2599858  764.4859813  0.3416333
GSU  FSPTS  RND/PCB 1093.654426   833.7446564  783.3823529  0.3401258
GSU  FSPTS  RND/PCB 1067.950072   814.8915352  772.705314  0.3362336
GSU  FSPTS  RND/PCB 1065.002818   811.7288995  767.3474178  0.3441099
GSU  FSPTS  SPT/PCB 1008.802136   786.4886172  769.8611111  0.3472745
GSU  FSPTS  SPT/PCB 1015.753339   807.7195226  763.5046729  0.3413855
GSU  FSPTS  SPT/PCB 1068.676277   851.2898738  783.0731707  0.3401655
GSU  FSPTS  SPT/PCB 1039.487998   839.9356044  769.8309179  0.3334803
GSU  FSPTS  SPT/PCB 1044.603026   854.9150049  767.5829384  0.3429601
;
proc anova;
  classes methode interf intraf;
  model wip = methode interf intraf methode*interf methode*intraf
             interf*intraf methode*interf*intraf;
  means methode/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;

```

```

run;
proc anova;
  classes methode interf intraf;
  model tpas = methode interf intraf methode*interf methode*intraf
    interf*intraf methode*interf*intraf;
  means methode/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;
run;
proc anova;
  classes methode interf intraf;
  model setup = methode interf intraf methode*interf methode*intraf
    interf*intraf methode*interf*intraf;
  means methode/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;
run;
proc anova;
  classes methode interf intraf;
  model ratio = methode interf intraf methode*interf methode*intraf
    interf*intraf methode*interf*intraf;
  means methode/duncan alpha=0.05;
  means interf/duncan alpha=0.05;
  means intraf/duncan alpha=0.05;
run;

```

Résultats de SAS:

1

```

Simulation Ligne SMT

Analysis of Variance Procedure
  Class Level Information

  Class      Levels      Values
  METHODE      2      GSU SDS
  INTERF       4      FSPT FSPTS MS RND
  INTRAF       4      RND RND/PCB SPT SPT/PCB

  Number of observations in data set = 160

```

—

2

```

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: WIP

  Source      DF      Sum of Squares      Mean Square      F Value      Pr > F
  Model       31      1053194.2127      33974.0069      46.89      0.0001

```

Error	128	92736.8715	724.5068	
Corrected Total	159	1145931.0841		
		R-Square	C.V.	Root MSE
		0.919073	3.704131	26.916664
				726.66607
Source	DF	Anova SS	Mean Square	F Value
METHODE	1	12022.33703	12022.33703	16.59
INTERF	3	974501.10162	324833.70054	448.35
INTRAF	3	43995.05357	14665.01786	20.24
METHODE*INTERF	3	19198.60318	6399.53439	8.83
METHODE*INTRAF	3	2832.10730	944.03577	1.30
INTERF*INTRAF	9	482.10012	53.56668	0.07
METHOD*INTERF*INTRAF	9	162.90987	18.10110	0.02
				0.0000

—

3

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 724.5068

Number of Means 2
Critical Range 8.421

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	735.334	80	GSU
B	717.998	80	SDS

—

4

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 724.5068

Number of Means	2	3	4
Critical Range	11.91	12.53	12.95

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	813.131	40	FSPT
B	795.709	40	FSPTS
C	652.654	40	RND
C	645.170	40	MS

-

5

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: WIP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 724.5068

Number of Means	2	3	4
Critical Range	11.91	12.53	12.95

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	744.569	40	SPT
A	741.814	40	SPT/PCB
B	711.511	40	RND/PCB
B	708.771	40	RND

-

6

Simulation Ligne SMT

Analysis of Variance Procedure Class Level Information

Class	Levels	Values
METHODE	2	GSU SDS
INTERF	4	FSPT FSPTS MS RND

INTRAF 4 RND RND/PCB SPT SPT/PCB

Number of observations in data set = 160

—

7

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: TPAS

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	3847460.1103	124111.6165	97.40	0.0001
Error	128	163099.0178	1274.2111		
Corrected Total	159	4010559.1281			
		R-Square	C.V.	Root MSE	TPAS Mean
		0.959333	2.816379	35.696093	1267.4462

Source

Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	820634.4384	820634.4384	644.03	0.0001
INTERF	3	2452723.6478	817574.5493	641.63	0.0001
INTRAF	3	159201.3014	53067.1005	41.65	0.0001
METHODE*INTERF	3	392929.7490	130976.5830	102.79	0.0001
METHODE*INTRAF	3	19497.5130	6499.1710	5.10	0.0023
INTERF*INTRAF	9	1780.4303	197.8256	0.16	0.9977
METHOD*INTERF*INTRAF	9	693.0304	77.0034	0.06	1.0000

—

8

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 1274.211

Number of Means 2
Critical Range 11.17

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	1339.063	80	SDS

B 1195.829 80 GSU

— 9

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 1274.211

Number of Means	2	3	4
Critical Range	15.79	16.62	17.17

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	1448.658	40	MS
B	1312.393	40	RND
C	1175.415	40	FSPTS
D	1133.318	40	FSPT

— 10

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: TPAS

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 1274.211

Number of Means	2	3	4
Critical Range	15.79	16.62	17.17

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	1317.255	40	RND
B	1263.386	40	RND/PCB
B	1259.340	40	SPT

C 1229.804 40 SPT/PCB

-

Simulation Ligne SMT

Analysis of Variance Procedure
Class Level Information

Class	Levels	Values
METHODE	2	GSU SDS
INTERF	4	FSPT FSPTS MS RND
INTRAF	4	RND RND/PCB SPT SPT/PCB

Number of observations in data set = 160

-

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: SETUP

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	2381311.6326	76816.5043	338.93	0.0001
Error	128	29010.3050	226.6430		
Corrected Total	159	2410321.9375			
		R-Square	C.V.	Root MSE	SETUP Mean
		0.987964	1.651857	15.054667	911.37832

Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	2148740.3867	2148740.3867	9480.73	0.0001
INTERF	3	0.0006	0.0002	0.00	1.0000
INTRAF	3	211512.4583	70504.1528	311.08	0.0001
METHODE*INTERF	3	0.0006	0.0002	0.00	1.0000
METHODE*INTRAF	3	21058.7832	7019.5944	30.97	0.0001
INTERF*INTRAF	9	0.0017	0.0002	0.00	1.0000
METHOD*INTERF*INTRAF	9	0.0017	0.0002	0.00	1.0000

-

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 226.643

Number of Means	2
Critical Range	4.710

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	1027.265	80	SDS
B	795.492	80	GSU

14

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 226.643

Number of Means	2	3	4
Critical Range	6.661	7.010	7.243

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	911.382	40	FSPTS
A	911.377	40	FSPT
A	911.377	40	MS
A	911.377	40	RND

15

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: SETUP

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 226.643

Number of Means	2	3	4
Critical Range	6.661	7.010	7.243

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	957.458	40	RND
B	936.484	40	SPT
C	876.282	40	RND/PCB
C	875.289	40	SPT/PCB

—

Simulation Ligne SMT

Analysis of Variance Procedure
Class Level Information

Class	Levels	Values
METHODE	2	GSU SDS
INTERF	4	FSPT FSPTS MS RND
INTRAF	4	RND RND/PCB SPT SPT/PCB

Number of observations in data set = 160

—

Simulation Ligne SMT

Analysis of Variance Procedure

Dependent Variable: RATIO

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	31	0.14806495	0.00477629	197.63	0.0001
Error	128	0.00309350	0.00002417		
Corrected Total	159	0.15115845			

	R-Square	C.V.	Root MSE	RATIO	Mean
	0.979535	1.307804	0.0049161	0.3759044	
Source	DF	Anova SS	Mean Square	F Value	Pr > F
METHODE	1	0.13818417	0.13818417	5717.65	0.0001
INTERF	3	0.00000000	0.00000000	0.00	1.0000
INTRAF	3	0.00913642	0.00304547	126.01	0.0001
METHODE*INTERF	3	0.00000000	0.00000000	0.00	1.0000
METHODE*INTRAF	3	0.00074435	0.00024812	10.27	0.0001
INTERF*INTRAF	9	0.00000000	0.00000000	0.00	1.0000
METHOD*INTERF*INTRAF	9	0.00000000	0.00000000	0.00	1.0000

-

18

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: RATIO

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 0.000024

Number of Means	2
Critical Range	.001538

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	METHODE
A	0.4052923	80	SDS
B	0.3465164	80	GSU

19

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: RATIO

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 0.000024

Number of Means	2	3	4
Critical Range	.002175	.002289	.002365

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTERF
A	0.375905	40	FSPTS
A	0.375904	40	FSPT
A	0.375904	40	MS
A	0.375904	40	RND

20

Simulation Ligne SMT

Analysis of Variance Procedure

Duncan's Multiple Range Test for variable: RATIO

NOTE: This test controls the type I comparisonwise error rate, not the experimentwise error rate

Alpha= 0.05 df= 128 MSE= 0.000024

Number of Means	2	3	4
Critical Range	.002175	.002289	.002365

Means with the same letter are not significantly different.

Duncan Grouping	Mean	N	INTRAF
A	0.386340	40	RND
B	0.379846	40	SPT
C	0.369277	40	RND/PCB
C	0.368155	40	SPT/PCB